



TUGAS AKHIR TF 145565

RANCANG BANGUN SISTEM PENGENDALIAN *pH* DALAM PROSES TRANSESTERIFIKASI PADA MINI PLAN BIODIESEL DI WORKSHOP INSTRUMENTASI

DIMAS GERRY AINUL WAFA'
NRP 2414.031.021

Dosen Pembimbing
Dr. Bambang Lelono Widjiantoro, S.T, M.T.
NIP. 19690507 199512 1 001

PROGRAM STUDI D3 TEKNIK INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017



FINAL PROJECT TF 145565

***DESIGN OF pH CONTROL SYSTEM DESIGN IN
TRANSESTERIFICATION PROCESS ON MINI PLAN
BIODIESEL IN WORKSHOP INSTRUMENTATION***

**DIMAS GERRY AINUL WAFA'
NRP 2414.031.021**

**ADVISOR LECTURER
Dr. Bambang Lelono Widjiantoro, ST, MT
NIP. 19690507 1995121 001**

***STUDY PROGRAM OF D3 METROLOGY AND INSTRUMENTATION
DEPARTMENT INSTRUMENTATION ENGINEERING
Faculty of Vocational
Sepuluh Nopember Institute of Technology
Surabaya 2017***



TUGAS AKHIR - TF 145565

RANCANG BANGUN SISTEM PENGENDALIAN *pH* DALAM PROSES TRANSESTERIFIKASI PADA MINI PLAN BIODIESEL DI WORKSHOP INSTRUMENTASI

DIMAS GERRY AINUL WAFA'
NRP 2414.031.021

Dosen Pembimbing
Dr. Bambang Lelono Widjiantoro, ST., MT.

PROGRAM STUDI D3 METROLOGI DAN INSTRUMENTASI
JURUSAN TEKNIK INSTRUMENTASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

**RANCANG BANGUN SISTEM PENGENDALIAN *pH*
DALAM PROSES TRANSESTERIFIKASI PADA
MINI PLAN BIODIESEL DI WORKSHOP
INSTRUMENTASI**

TUGAS AKHIR

Oleh :

DIMAS GERRY AINUL WAFA
NRP. 2414 031 021

Surabaya, 28 Juli 2017
Mengetahui / Menyetujui

Dosen Pembimbing



Dr. Bambang Lelono Widjiantoro, ST, MT
NIP. 19690507 1995121 001

Kepala Departemen
Teknik Instrumentasi TV – ITS



Dr. Ir. Purwadi Agus D., M.Sc
NIP. 19620822 198803 1 001

**"RANCANG BANGUN SISTEM PENGENDALIAN *pH*
DALAM PROSES TRANSESTERIFIKASI PADA
MINI PLAN BIODIESEL DI WORKSHOP
INSTRUMENTASI"**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
Pada
Program Studi D3 Teknik Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

DIM AS GERRY AINUL WAFA
NRP. 2414 031 021

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Bambang Lelono W. S.T., M.T. (Dosen Pembimbing)
2. Ir. Tutug Dhanardono M.T. Ketua Tim Penguji)
3. Ir. Yaumar M.T. (Dosen Penguji 1)

SURABAYA
JULI 2017

“RANCANG BANGUN SISTEM PENGENDALIAN *pH* DALAM PROSES TRANSESTERIFIKASI PADA *MINI* *PLANT* BIODIESEL DI WORKSHOP INSTRUMENTASI”

Nama : Dimas Gerry Ainul Wafa
NRP : 2414 031 021
Program Studi : D3 Metrologi dan Instrumentasi
Dosen Pembimbing : Dr. Bambang Lelono W., ST, MT

Abstrak

Pada sistem pengendalian *ph* ini merupakan suatu rancangan sistem yang mampu mengendalikan *ph* pada *mini plant biodiesel process* khususnya pada tangki pencampuran minyak dan larutan metoksida yang terdiri dari *methanol* (CH_3OH) dan katalis (KOH). Pengendalian *ph* pada proses transesterifikasi *mini plant biodiesel* menggunakan sistem pengendalian bermode *on-off* dan menggunakan sensor *ph probe* sebagai sensor *ph probe*, Atmega 8535 sebagai *controller*, LCD yang berfungsi untuk menampilkan data dan aktuatur yaitu *solenoid valve*. Cara kerja dari plan pengendalian ini yaitu sensor *ph* akan mendeteksi perubahan *ph* pada tangki pencampuran kemudian diolah oleh mikrokontroler Atmega 8535 sehingga perubahan *ph* dapat ditampilkan melalui LCD. Pada tangki ini *ph* ideal pencampuran minyak dan metoksida adalah 8-9, maka data yang diambil juga berbeda dengan set point yaitu 8 dan 9. Pada hasil pengujian dapat terlihat bahwa *ph* optimum yang digunakan pada proses transesterifikasi adalah pada set point 9 karena dapat dapat menghasilkan pencampuran larutan metoksida dan minyak dengan baik. Pada sistem pengendalian temperatur tangki pencampuran mempunyai nilai akurasi sebesar 96,90 %.

Kata kunci : Sistem pengendalian *ph*, proses transesterifikasi, sensor *ph probe* df robot, mikrokontroler Atmega 8535

“A DESIGN OF pH CONTROL SYSTEM IN THE TRANSESTERIFICATION PROCESS TO BIODIESEL MINI PLANT IN THE INSTRUMENTATION WORKSHOP”

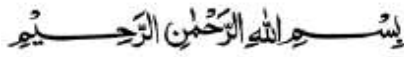
Name : Dimas Gerry Ainul Wafa
NRP : 2414 031 021
Department : Metrology and Instrumentation
Engineering Diploma Program
Advisor Lecturer : Dr. Bambang lelono W., ST, MT

Abstract

The ph control system is a system that can control the ph of biodiesel mini plant process, especially in the mixing tank of oil and metoxide that consist of methanol (CH_3OH) and catalyst (KOH). The ph control system in the transesterification process of biodiesel mini plant is using on-off control mode with ph probe as ph sensor, atmega 8535 as a controller, LCD is used to display data and actuator is solenoid valve. The responsible of the control plant is the ph probe sensor will detect the ph changes on the mixing tank then processed by Atmega 8535 microcontroller, so that the ph changes can be displayed through LCD. On this tank the ideal ph of mixing oil and metoxide is 8- 9, so that the data which is taken are also different with the set point is 8, 9. On the result of the test can be seen that the optimum ph is on the set point of 9 that is used on the transesterification process because it can produce good mixing from metoxide and the oil. The calculation result of accuration system is 96,90 %.

Keywords : *The ph control system, The transesterification process, The ph probe sensor, The Atmega 8535 microcontroller*

KATA PENGANTAR



Puji syukur Alhamdulillah kami panjatkan kehadiran Allah SWT serta Nabi Muhammad SAW atas berkah, limpahan rahmat dan hidayah-Nya kepada kami semua sehingga penulis dapat menyelesaikan Tugas Akhir tepat waktu dengan judul :

“RANCANG BANGUN SISTEM PENGENDALIAN *PH* DALAM PROSES TRANSESTERIFIKASI PADA MINI PLANT BIODIESEL PROCESS DI WORKSHOP INSTRUMENTASI”

Tugas akhir ini disusun guna memenuhi persyaratan untuk memperoleh gelar Diploma pada departemen Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama menyelesaikan tugas akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Ir.Purwadi Agus Darwito,M.Sc selaku Kepala Departemen Teknik Instrumentasi yang telah memberikan dukungan dan semangat kepada kami.
2. Bapak Dr. Bambang Lelono Widjiantoro, ST, MT ,selaku dosen wali yang selalu memberikan motivasi dan perhatiannya selama penulis menjadi mahasiswa di Jurusan Teknik Instrumentasi
3. Bapak Dr. Ir. Totok Soehartanto, DEA selaku Kepala Laboratorium Workshop Instrumentasi yang telah memberikan izin menggunakan *workshop* intrumentasi sebagai tempat pengerjaan tugas akhir kami.
4. Orang Tua saya yang tercinta,terutama Ibu saya, terima kasih atas segala dukungan dan kepercayaan baik moril, spiritual dan material. Semoga selalu dilimpahkan rahmat dan hidayahnya.

5. Kelompok TA Biodiesel Proses Lintang gumebyar, Renza yuarifaldi, Andi fathurahman, Atika diyanasari, Anak Agung Yuniartha yang bersama-sama berjuang dalam pengerjaan tugas akhir. Semoga akan selalu teringat kebersamaan kita.
6. Teman-teman seperjuangan tugas akhir D3 Metrologi dan Instrumentasi angkatan 2014 yang tidak dapat disebutkan satu persatu dan semua pihak yang telah membantu terselesaikannya Tugas Akhir ini.
7. Kakak-kakak dari lintas jalur maupun alumni yang selalu memberi bimbingan dan pengetahuan seputar tugas akhir.
8. Serta semua pihak yang turut membantu terselesaikannya Tugas Akhir ini, terima kasih banyak

Penulis menyadari bahwa terdapat kekurangan dalam penyusunan laporan tugas akhir ini. Karena itu sangat diharapkan kritik dan saran yang membangun dari semua pihak sehingga mencapai sesuatu yang lebih baik.

Surabaya, 20 Juli 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Metodologi Pengujian Alat	4
1.6 Sistematika Laporan	4
 BAB II TEORI PENUNJANG	 7
2.1 Bahan Baku Biodiesel	7
2.2 <i>Mixing Process</i>	9
2.3 Sensor <i>pH probe meter</i>	11
2.4 Reaksi Transesterifikasi	12
2.5 <i>Solenoid Valve</i>	13
2.6 Microcontroller Atmega 8535	14
2.7 Sistem Pengendalian <i>on/off</i>	21
2.8 Lampu Indikator	23
2.9 LCD (<i>Liquid Cristal Display</i>)	24
2.10 <i>Relay</i>	26
 BAB III PERANCANGAN DAN PEMBUATAN ALAT	 29
3.1 Blok Diagram Perancangan Alat	29
3.2 Perancangan Sistem Pengendalian <i>PH</i> Pada <i>Unit</i> <i>Mixing</i>	30
3.3 Perancangan <i>Local Control unit (LCU)</i>	32

3.4	Perancangan Hardware	33
3.5	Perancangan Perangkat Lunak <i>Software</i>	39
3.6	Prosedur Operasional	41
 BAB IV PENGUJIAN ALAT DAN ANALISA DATA		43
4.1	Pengujian Alat	43
4.2	Pengujian Sistem	43
4.3	Analisa Data.....	54
 BAB V PENUTUP		57
5.1	Kesimpulan	57
5.2	Saran	57
 DAFTAR PUSTAKA		
LAMPIRAN A DATA SHEET ATMEGA 8535		
MICROCONTROLLER		
LAMPIRAN B DATA SHEET SENSOR PH PROBE DF		
ROBOT		
LAMPIRAN C PEMROGRAMAN PH DENGAN ATMEGA		
8535		

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Mixing tank</i>	10
Gambar 2.2 <i>Solenoid Valve</i>	13
Gambar 2.3 <i>Bagian-Bagian Solenoid Valve</i>	14
Gambar 2.4 <i>Konfigurasi pin IC ATmega 8535</i>	17
Gambar 2.5 <i>Fluktuasi Pengendalian On-Off</i>	23
Gambar 2.6 <i>Lampu Indikator</i>	24
Gambar 2.7 <i>Konfigurasi LCD 2x16</i>	24
Gambar 2.8 <i>Relay DC</i>	27
Gambar 3.1 <i>Blok Diagram Perancangan dan Pembuatan Alat</i>	29
Gambar 3.2 <i>Rancangan Plant Temperatur Pada Tangki Pencampuran</i>	30
Gambar 3.3 <i>Solenoid Valve</i>	31
Gambar 3.4 <i>Sensor ph probe df robot</i>	31
Gambar 3.5 <i>Diagram Blok Pengendalian ph</i>	30
Gambar 3.6 <i>LCU plant Pengendalian ph</i>	33
Gambar 3.7 <i>Rangkaian Power Supply 5 Volt</i>	34
Gambar 3.8 <i>Rangkaian Power Supply 12 Volt</i>	34
Gambar 3.9 <i>Penempatan Sensor ph probe df robot</i>	35
Gambar 3.10 <i>Rangkaian Modul Relay</i>	36
Gambar 3.11 <i>Koneksi LCD 2x16 ke Microcontroller</i>	37
Gambar 3.12 <i>Tampilan Perancangan LCD Pada Uji Coba Sensor ph probe df robot</i>	37
Gambar 3.13 <i>solenoid zero pressure</i>	38
Gambar 3.14 <i>solenoid dengan pressure</i>	39
Gambar 3.15 <i>Diagram Alir Program</i>	40
Gambar 4.1 <i>Grafik Perubahan Temperatur Terhadap Waktu Pada Set Point ph 9</i>	46
Gambar 4.2 <i>Grafik Pengujian Sistem pada set point ph 8</i>	48
Gambar 4.3 <i>Grafik Pengujian Sistem pada set point ph 9</i>	52
Gambar 4.4 <i>Grafik Respon Sistem Pengendalian pH</i>	54
Gambar 4.5 <i>Diagram blok pengendalian pH</i>	54

DAFTAR TABEL

	Halaman
Tabel 2.1 Konfigurasi <i>Port B</i>	18
Tabel 2.2 Konfigurasi <i>Port D</i>	19
Tabel 2.3 Konfigurasi LCD 2x16	25
Tabel 4.1 Pengukuran ph Pada Set Point ph 9	44
Tabel 4.2 Data Pengujian Sistem dengan <i>set point</i> ph 8	47
Tabel 4.3 Tabel Data <i>Error (ess)</i> dengan <i>Set Point</i> ph 8.....	49
Tabel 4.4 Data Pengujian Sistem dengan <i>set point ph</i> 9	50
Tabel 4.5 Tabel Data <i>Error (ess)</i> dengan <i>set point</i> ph 9	52
Tabel 4.6 Data Perhitungan Akurasi	53

BAB I

PENDAHULUAN

1.1 Latar Belakang

Krisis energi dunia yang terjadi pada dekade terakhir memberikan dampak yang signifikan pada meningkatnya harga bahan bakar minyak (BBM), telah mendorong pengembangan energi alternatif dengan pemanfaatan sumberdaya energi terbarukan (*renewable resources*). Salah satu bentuk energi alternatif yang saat ini mulai dikembangkan adalah *biodiesel* yang mempunyai tingkat kelayakan teknologi cukup tinggi. Untuk mendorong pengembangan *biodiesel*, pemerintah telah mengeluarkan Kebijakan Energi Nasional diantaranya dengan menetapkan target produksi *biodiesel* pada tahun 2025 sebesar 5 % dari total kebutuhan energi minyak nasional dan penugasan kepada Departemen Kehutanan untuk berperan dalam penyediaan bahan baku *biodiesel* termasuk pemberian ijin pemanfaatan lahan hutan terutama lahan yang tidak produktif. Salah satu tanaman hutan yang mempunyai potensi sebagai bahan baku *biodiesel* adalah Nyamplung (*Calophyllum inophyllum* L.). Nyamplung termasuk dalam marga *Callophyllum* yang mempunyai sebaran cukup luas di dunia yaitu Madagaskar, Afrika Timur, Asia Selatan dan Tenggara, Kepulauan Pasifik, Hindia Barat, dan Amerika Selatan. Di Indonesia nyamplung tersebar mulai dari bagian Barat sampai Bagian Timur Indonesia. Distribusi pohon nyamplung di Indonesia, mulai Sumatera Barat, Riau, Jambi, Sumatera Selatan, Lampung, Jawa, Kalimantan Barat, Kalimantan Tengah, Sulawesi, Maluku, hingga Nusa Tenggara Timur dan Papua . Selain itu, pohon tersebut juga ditemui di wilayah Malaysia, Filipina, Thailand, dan Papua Nugini. Kelebihan nyamplung sebagai bahan baku *biodiesel* adalah biji mempunyai rendemen yang tinggi (bisa mencapai 74%) dan dalam pemanfaatannya tidak berkompetisi dengan kepentingan pangan. Selain itu, nyamplung memiliki keunggulan ditinjau dari prospek pengembangan dan pemanfaatan lain, antara lain : tumbuh dan tersebar merata secara alami di Indonesia, regenerasi

mudah dan berbuah sepanjang tahun menunjukkan daya *survival* yang tinggi terhadap lingkungan, relatif mudah dibudidayakan baik tanaman sejenis (*monoculture*) atau hutan campuran (*mixed forest*), cocok di daerah beriklim kering, permudaan alami banyak, dan berbuah sepanjang tahun, hampir seluruh bagian tanaman nyamplung berdayaguna dan menghasilkan bermacam produk yang memiliki nilai ekonomi, tegakan hutan nyamplung berfungsi sebagai *wind breaker*/ perlindungan untuk tanaman pertanian dan konservasi sempa dan pantai, pemanfaatan *biodiesel* nyamplung dapat menekan laju penebangan pohon hutan sebagai kayu bakar. (Santoso.,2008) Pada pembuatan biodiesel, bahan baku mengalami tahap pemurnian (*refining*) terlebih dahulu sebelum dilakukan transesterifikasi. Tahap pemurnian ini bertujuan untuk menghilangkan berbagai bahan pengotor yang terkandung di dalam minyak, seperti air, asam lemak bebas, gliserida parsial, fosfatida, produk oksidasi, pigmen, lilin, dan elemen-elemen lainnya. Biodiesel merupakan bahan bakar alternatif yang tepat dari bahan mentah terbaharukan (*renewable*) selain bahan bakar diesel dari minyak bumi. Dibandingkan bahan bakar fosil, biodiesel mempunyai kelebihan diantaranya bersifat dapat terdegradasi dengan mudah (*biodegradable*), *non-toxic*, mempunyai angka emisi dan gas sulfur yang rendah sehingga emisi pembakaran yang dihasilkan ramah lingkungan serta tidak menambah akumulasi gas karbondioksida di atmosfer sehingga lebih mengurangi efek pemanasan global atau banyak disebut dengan *zero CO₂ emission*. Dari berbagai jenis tanaman dan limbah yang dapat dijadikan sebagai sumber bahan baku biodiesel, salah satunya adalah dengan memanfaatkan limbah biji randu/kapuk yang memiliki kandungan 18-25% minyak, sehingga dapat diolah menjadi biodiesel. Selain itu, keberadaan biji randu yang cukup melimpah merupakan kesempatan besar untuk bisa dikembangkan menjadi pilihan energi alternatif dalam skala komersial karena minyak biji randu selama ini hanya digunakan sebagai bahan baku alat penerangan, minyak pelumas, campuran *coating* pada genting,

campuran pada kain batik, serta digunakan sebagai makanan ternak [2].

Maka dari itu akan dirancang sebuah *mini plant Biodiesel Process* secara sederhana. Sistem pada *mini plant Biodiesel Process* ini memerlukan sistem pengendalian otomatis bermode *on-off* ini berfungsi untuk mengendalikan PH pada *unit mixing* dengan campuran minyak dan metoksida (*Methanol* dan *KOH*) agar mendapatkan kualitas pencampuran yang baik. Oleh karena itu, akan dirancang suatu sistem pengendalian otomatis dengan menggunakan Atmega 8535 dengan bermode pengendalian *on-off* sebagai *controller* dan menggunakan sensor PH probe meter sebagai pendeteksi PH pada proses transesterifikasi *mini plant Biodiesel*.

1.2 Perumusan Masalah

Pada pelaksanaan tugas akhir ini terdapat permasalahan yaitu bagaimana merancang dan membangun sistem *control* PH pada tangki pencampuran ***Mini Plant Biodiesel Process***, agar pencampuran (*mixing*) antara minyak dan larutan campuran katalis (*KOH*) dan *methanol* dapat menghasilkan percepatan reaksi yang baik dan menjaga kestabilan temperatur saat pencampuran agar berlangsung dengan baik.

1.3 Batasan Masalah

Perlu diberikan beberapa batasan permasalahan agar pembahasan tidak meluas dan menyimpang dari tujuan. Adapun batasan permasalahan dari sistem yang dirancang ini yaitu :

- Pada *plant* tangki pencampuran, variabel proses yang akan dikendalikan adalah PH.
- Bahan yang digunakan dalam pembuatan biodiesel ini adalah minyak biji nyamplung, larutan alkohol (CH_3OH), katalis (*KOH*).
- PH yang digunakan dalam proses pencampuran ini
- Mode pengendali yang digunakan adalah *on-off*.
- Sensor yang digunakan yaitu PH dengan *range* pengukuran
- *Controller* yang digunakan adalah Atmega 8535

- Dilakukan *monitoring* terhadap perubahan PH.

1.4 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini yaitu merancang dan membangun suatu sistem pengendalian PH pada proses transesterifikasi **Mini Plant Biodiesel**, untuk mendapatkan hasil pencampuran dan konversi yang optimum.

1.5 Metodologi Pengujian Alat

Metodologi penelitian dalam pelaksanaan tugas akhir ini, yaitu :

- Studi literatur
Studi literatur meliputi pemahaman secara teori mengenai Atmega 8535, sensor PH probe meter dan teori tentang sistem pengendalian PH bermode *on-off*.
- Perancangan dan Pembuatan Alat
Merancang mekanik, mulai dari penempatan tangki minyak dan larutan campuran (KOH dan *methanol*) dan tangki pencampuran dan *solenoid valve* yang akan dipakai untuk membuka aliran minyak dan larutan campuran (KOH dan *methanol*). Merancang elektrik, mulai dari menghubungkan sensor PH probe meter menuju Atmega yang dihubungkan dengan LCD. Merancang *software* dengan menggunakan pemrograman bahasa C yang akan digunakan untuk memprogram Atmega 8535.
- Pengujian dan Analisa Data
Pengujian alat meliputi pengujian *hardware* dan pengujian *software* dengan cara mengoperasikan dan menjalankan *hardware* dan *software* tersebut dan hasilnya sesuai dengan perencanaan alat.

1.6 Sistematika Laporan

Dalam penyusunan tugas akhir ini, sistematika laporan disusun secara sistematis dan terbagi dalam beberapa bab, yaitu dengan perincian sebagai berikut:

BAB I**Pendahuluan**

Bab ini berisi tentang penjelasan latar belakang, perumusan masalah, batasan masalah, tujuan, tugas akhir, metodologi penelitian dan sistematika laporan.

BAB II**Tinjauan Pustaka**

Pada bab ini berisikan tentang teori-teori penunjang tugas akhir, antara lain teori tentang bahan pembuatan biodiesel antara lain minyak biji randu/kapuk, larutan *methanol* (CH_3OH) dan katalis (KOH), sensor PH probe meter, *relay*, cara kerja *solenoid valve*, proses *unit mixing*, reaksi transesterifikasi, *heater*, sistem pengendalian *on-off*, lampu indikator, LCD, dan Atmega 8535 *microcontroller*.

BAB III**Perancangan dan Pembuatan Alat**

Dalam bab ini akan dijelaskan secara detail mengenai langkah-langkah yang harus dilalui untuk mencapai tujuan dan simpulan akhir dari penelitian. Produk akhir dari tahap ini adalah perancangan dan model yang siap untuk dibuat, diuji, dan dianalisa.

BAB IV**Pengujian Alat dan Analisa Data**

Pada bab ini merupakan tindak lanjut dari bab III, dimana pengujian yang telah dilakukan dan akan didapatkan data, baik data berupa grafik maupun tabulasi, kemudian akan dilakukan analisa dan pembahasan.

BAB V**Kesimpulan dan Saran**

Dalam bab ini adalah berisi mengenai kesimpulan pokok dari keseluruhan rangkaian penelitian yang telah dilakukan serta saran yang dapat dijadikan rekomendasi sebagai pengembangan penelitian selanjutnya.

“Halaman ini sengaja dikosongkan”

BAB II

TEORI PENUNJANG

Dalam pengerjaan tugas akhir adapun teori penunjang yang digunakan sehingga tugas akhir ini dapat terselesaikan antara lain : komponen *mini plant* sistem pengendalian PH pada bahan pembuatan biodiesel yaitu minyak biji nyamplung, larutan *methanol* (CH_3OH), dan katalis (KOH), *mixing process*, reaksi transesterifikasi, *sensor* PH probe meter, *solenoid valve*, Atmega 8535, pengendalian *On-Off*, LCD, lampu indikator, *relay*.

2.1 Bahan Baku Biodiesel

Untuk membuat biodiesel diperlukan tiga komponen utama yaitu minyak biji kapuk, alkohol, dan katalis.

2.1.1 Minyak Biji Nyamplung

Nyamplung (*Calophyllum inophyllum* L.). Nyamplung termasuk dalam marga *Callophylum* yang mempunyai sebaran cukup luas di dunia yaitu Madagaskar, Afrika Timur, Asia Selatan dan Tenggara, Kepulauan Pasifik, Hindia Barat, dan Amerika Selatan. Di Indonesia nyamplung tersebar mulai dari bagian Barat sampai Bagian Timur Indonesia. Distribusi pohon nyamplung di Indonesia, mulai Sumatera Barat, Riau, Jambi, Sumatera Selatan, Lampung, Jawa, Kalimantan Barat, Kalimantan Tengah, Sulawesi, Maluku, hingga Nusa Tenggara Timur dan Papua . Selain itu, pohon tersebut juga ditemui di wilayah Malaysia, Filipina, Thailand, dan Papua Nugini. Kelebihan nyamplung sebagai bahan baku biodiesel adalah biji mempunyai rendemen yang tinggi (bisa mencapai 74%) dan dalam pemanfaatannya tidak berkompetisi dengan kepentingan pangan. Selain itu, nyamplung memiliki keunggulan ditinjau dari prospek pengembangan dan pemanfaatan lain, antara lain : tumbuh dan tersebar merata secara alami di Indonesia, regenerasi mudah dan berbuah sepanjang tahun menunjukkan daya survival yang tinggi terhadap lingkungan, relatif mudah dibudidayakan baik tanaman sejenis (*monoculture*) atau hutan campuran (*mixed*

forest), cocok di daerah beriklim kering, permudaan alami banyak, dan berbuah sepanjang tahun, hampir seluruh bagian tanaman nyamplung berdayaguna dan menghasilkan bermacam produk yang memiliki nilai ekonomi, tegakan hutan nyamplung berfungsi sebagai wind breaker/ perlindungan untuk tanaman pertanian dan konservasi sempa dan pantai, pemanfaatan biodiesel nyamplung dapat menekan laju penebangan pohon hutan sebagai kayu bakar. (Santoso.,2008)

2.1.2 Alkohol (*Methanol*)

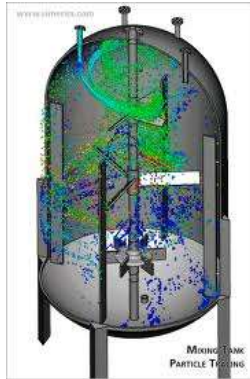
Jenis alkohol yang biasanya dipakai pada proses pembuatan biodiesel adalah *methanol* dan etanol. *Methanol* atau biasa dikenal sebagai metil alkohol merupakan jenis alkohol yang biasa digunakan dalam pembuatan biodiesel karena *methanol* (CH_3OH) mempunyai keuntungan lebih mudah bereaksi atau lebih stabil dibandingkan dengan etanol ($\text{C}_2\text{H}_5\text{OH}$) karena *methanol* memiliki satu ikatan carbon sedangkan etanol memiliki dua ikatan carbon, sehingga lebih mudah memperoleh pemisahan gliserol dibanding dengan etanol. Kerugian dari *methanol* adalah *methanol* termasuk zat yang beracun dan berbahaya, bahkan lebih mudah terbakar bila dibandingkan bensin. *Methanol* berwarna bening seperti air, mudah menguap, mudah terbakar dan mudah bercampur dengan air. Etanol lebih aman, tidak beracun dan terbuat dari hasil pertanian, sedangkan *methanol* mengandung uap yang berbahaya bagi makhluk hidup dan terbuat dari batubara. Etanol memiliki sifat yang sama dengan metanol yaitu berwarna bening seperti air, mudah menguap, mudah terbakar dan mudah bercampur dengan air. Namun, saat ini penggunaan *methanol* lebih banyak digunakan karena selain harganya lebih murah, juga lebih mudah untuk memperoleh pemisahan gliserin dibandingkan dengan etanol, dan pada penggunaan etanol apabila tidak berhati-hati akan menimbulkan emulsi. *Methanol* dan etanol yang dapat digunakan hanya yang murni 99%. *Methanol* memiliki massa jenis 0,7915 g/m³, sedangkan etanol memiliki massa jenis 0,79 g/m³ [5].

2.1.3 Katalis (KOH)

Katalis adalah suatu zat yang berfungsi mempercepat laju reaksi dengan menurunkan energi aktivasi, namun tidak menggeser letak keseimbangan. Tanpa katalis reaksi transesterifikasi akan berjalan sangat lambat karena membutuhkan temperatur dan tekanan yang tinggi untuk mencapai hasil yang maksimum. Katalis dapat berupa basa, asam, atau enzim. Namun, katalis basa lebih banyak digunakan sebagai katalis dalam reaksi transesterifikasi karena menghasilkan metil ester yang tinggi dan waktu yang cepat. Katalis yang digunakan adalah natrium hidroksida (NaOH) atau kalium hidroksida (KOH dengan sifat basa kuat yang dimiliki). Katalis akan memecahkan minyak biji kapuk dan melepaskan ester, begitu ester bebas akan menempel pada alkohol. Sedangkan katalis dan gliserol akan mengendap. Jumlah katalis yang digunakan harus tepat. Penggunaan katalis yang terlalu sedikit akan menyebabkan minyak dan alkohol tidak bereaksi, dan apabila jumlah katalis yang digunakan terlalu banyak akan menyebabkan campuran terelmusi ^[6].

2.2 *Mixing Process*

Mixing process merupakan suatu proses membaurkan beberapa bahan menjadi satu kesatuan dengan presentase tertentu sehingga menghasilkan produk baru yang homogen. Contoh dari proses pencampuran cair-cair yaitu seperti sirup, bahan-bahan kimia, minuman kaleng dan sebagainya. Dalam proses tersebut diperlukan gaya mekanik sebagai penggerak alat pengaduk sehingga proses pencampuran dapat berjalan dengan baik. Ada beberapa metode dalam proses pencampuran seperti penyemprotan dan mengeluarkan gas ke dalam cairan. Namun metode yang paling sering digunakan yaitu dengan metode turbulensi dimana bahan-bahan yang akan dicampur dimasukkan kedalam bejana kemudian diaduk^[7].



Gambar 2.1. *Mixing Tank* ^[7]

Selain itu salah satu hal yang perlu diperhatikan dalam menentukan peralatan yang harus digunakan dalam pencampuran adalah fase dari bahan yang akan dicampur. Berikut akan dibahas pencampuran berdasarkan fase dari bahan yang dicampur.

1. *Liquid-Liquid Material of Mixing*

Mixing process ini digunakan untuk mempersiapkan atau melangsungkan proses-proses kimia dan fisika serta juga untuk membuat produk akhir yang komersial. Alat yang digunakan untuk mencampur bahan cair-cair dapat berupa tangki atau bejana yang dilengkapi dengan pengaduk. Tangki atau bejana biasanya berbentuk silinder dengan sumbu terpasang vertikal, bagian atas bejana itu bias terbuka saja ke udara atau dapat pula tertutup. Ujung bawah tangki itu biasanya agak membulat, jadi tidak datar saja, maksudnya agar tidak terdapat terlalu banyak sudut-sudut tajam atau daerah yang sulit ditembus arus zat cair. Kedalaman zat cair biasanya hampir sama dengan diameter tangki. Di dalam tangki itu dipasang pengaduk (*impeller*) pada ujung poros menggantung, artinya poros itu ditumpu dari atas. Poros itu digerakkan oleh motor, yang kadang-kadang dihubungkan langsung dengan poros itu, namun biasanya dihubungkan melalui peti roda gigi untuk menurunkan kecepatannya.

2. *Solid-liquid Material of Mixing*

Pada proses pembuatan produk industri kimia yang siap

untuk diperdagangkan dan pada pengolahan produk setengah jadi, seringkali bahan-bahan padat harus dicampurkan dengan sejumlah kecil cairan. Di sini dapat terbentuk bahan padat yang lembab atau campuran yang sangat viskos seperti pasta atau adonan. Seringkali cairan harus juga ditambahkan ke dalam pasta, adonan atau massa yang plastis tersebut. Alat yang digunakan dapat berupa tangki atau bejana vertikal yang berbentuk silinder, bahan digilas dan diuli oleh satu atau dua perkakas campur yang mirip pengaduk.

3. *Solid-solid Material of Mixing*

Mixing Process dengan dua atau lebih dari bahan padat banyak dijumpai yang akan menghasilkan produk komersial industri kimia. Alat yang digunakan untuk pencampuran bahan padat dengan padat dapat berupa bejana-bejana yang berputar, atau bejana-bejana berkedudukan tetap tapi mempunyai perlengkapan pencampur yang berputar, ataupun pneumatik.

4. *Liquid-gas Material of Mixing*

Untuk *mixing process* kimia dan fisika tertentu gas harus dimasukkan ke dalam cairan, artinya cairan dicampur secara sempurna dengan bahan-bahan berbentuk gas.

2.3 Sensor pH Probe meter

pH meter adalah sebuah alat elektronik yang berfungsi untuk mengukur pH (derajat keasaman atau kebasaan) suatu cairan (ada elektroda khusus yang berfungsi untuk mengukur pH bahan-bahan semi-padat). Sebuah pH meter terdiri dari sebuah elektroda (*probe* pengukur) yang terhubung ke sebuah alat elektronik yang mengukur dan menampilkan nilai pH. alat ini sangat berguna untuk industri air minum, laboratorium, akuarium, industri pakaian terutama batik dan pewarna pakaian. Sensor yang biasa digunakan untuk mengukur pH adalah elektroda yang sensitif terhadap ion atau disebut juga elektroda gelas. Elektroda ini tersusun dari batang elektroda (terbuat dari gelas yang terisolasi dengan baik) dan membran gelas (yang berdinding tipis dan sensitif terhadap ion H^+). Elemen sensor pengukur pH terdapat di tengah-tengah, dilingkupi oleh larutan perak-perak klorida

(Ag-AgCl). Bagian bawah dari elemen sensor ini berhubungan dengan membran gelas dan berisi larutan perak-perak klorida. Kontak ionik dari larutan perak-perak klorida terhadap sampel terjadi melalui penghubung keramik. Penghubung ini bertindak sebagai suatu membran selektif yang hanya meloloskan arus-arus ionik tertentu, Secara alami, impedansi keluaran elektroda gelas sangat besar (karena proses kimia yang terjadi pada permukaan elektroda), besarnya antara 50-500 M Ω sehingga pada alat pengukur diperlukan impedansi masukan yang sangat besar (Coughlin, 1994). Pada prinsipnya pengukuran suatu pH adalah didasarkan pada potensial elektro kimia yang terjadi antara larutan yang terdapat didalam elektroda gelas (membrane gelas) yang telah diketahui dengan larutan yang terdapat diluar elektroda gelas yang tidak diketahui. Hal ini dikarenakan lapisan tipis dari gelembung kaca akan berinteraksi dengan ion hidrogen yang ukurannya relatif kecil dan aktif, elektroda gelas tersebut akan mengukur potensial elektrokimia dari ion hidrogen atau diistilahkan dengan potential of hidrogen. Untuk melengkapi sirkuit elektrik dibutuhkan suatu elektroda pembanding. Sebagai catatan, alat tersebut tidak mengukur arus tetapi hanya mengukur tegangan.

2.4 Reaksi Transesterifikasi

Salah satu proses yang harus dilakukan untuk memproduksi biodiesel adalah dengan cara reaksi transesterifikasi. Reaksi transesterifikasi merupakan reaksi katalisa kimia yang melibatkan minyak biji kapuk dan alkohol untuk mendapatkan asam lemak alkil ester (biodiesel) dan gliserol sebagai produk samping. Pada saat ini sebagian produksi biodiesel menggunakan *methanol* sebagai alkohol yang paling umum digunakan, karena harganya yang murah dan reaktifitasnya tinggi. Transesterifikasi juga menggunakan katalis dalam reaksinya. Tanpa adanya katalis, konversi yang dihasilkan maksimum namun reaksi akan berjalan dengan lambat. Katalis yang digunakan pada proses transesterifikasi adalah katalis basa seperti natrium hidroksida (NaOH) atau kalium hidroksida (KOH), karena katalis ini dapat mempercepat reaksi ^[8].

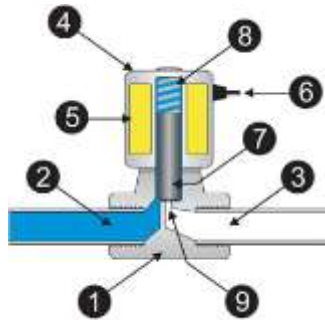
2.5 Solenoid Valve

Solenoid Valve merupakan sebuah katup yang mempunyai kumparan sebagai penggerak piston yang digerakan dengan arus AC maupun DC. Pada solenoid valve terdapat lubang keluaran (*ouput*), lubang masukan (*input*) dan *exhaust*. Masing-masing lubang tersebut mempunyai fungsi yang berbeda, misalnya lubang keluaran berfungsi sebagai jalan masuk *fluida* atau *supply*, kemudian lubang keluaran berfungsi sebagai jalan tempat keluarnya *fluida* dan *exhaust* sebagai tempat mengeluarkan cairan yang terjebak saat piston bergerak atau pindah posisi saat *solenoid* bekerja.



Gambar 2.2. *Solenoid Valve* ^[12]

Solenoid valve bekerja ketika katup listrik yang mempunyai koil sebagai penggerak dimana ketika koil mendapat *supply* tegangan listrik maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakkan piston pada bagian dalam. Ketika piston berpindah posisi maka pada lubang keluaran solenoid valve akan mengeluarkan fluida sesuai dengan *supply* tegangan yang diterima, umumnya 100/200 VAC tetapi ada juga yang mendapatkan *supply* tegangan DC. Berikut ini merupakan bagian-bagian dari solenoid valve:



Gambar 2.3. Bagian-Bagian *Solenoid Valve* ^[12]

Keterangan:

1. *Valve body*
2. Terminal masukan (*inlet port*)
3. Terminal keluaran (*outlet port*)
4. Koil / koil *solenoid*
5. Kumparan
6. Kabel suplai tegangan
7. *Plunger*
8. *Spring*
9. Lubang / *exhaust*

2.6 Mikrokontroler Atmega 8535

Mikrokontroler merupakan suatu rangkaian terintegrasi (*IC*) dengan kepadatan yang sangat tinggi dan semua bagian yang diperlukan oleh suatu kontroler sudah dikemas dalam satu keping yang didalamnya terdiri dari pusat pemroses (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM / EPROM / PROM, unit *input/output*, antarmuka serial dan *parallel*, *timer* dan *counter*, serta *interrupt* kontroler. Mikrokontroler tersedia dalam beberapa pilihan, tergantung dari keperluan dan kemampuan yang diinginkan. Mikrokontroler yang banyak beredar biasanya terdiri dari 4, 8, 16 atau 32 bit.

Mikrokontroler AVR memiliki arsitektur RISC 8 Bit, sehingga semua instruksi dikemas dalam kode 16-bit (16-bits *word*) dan sebagian besar instruksi dieksekusi dalam satu siklus instruksi *clock*. Dan ini sangat membedakan sekali dengan instruksi MCS-51 (Berarsitektur CISC) yang membutuhkan siklus 12 *clock*. RISC adalah *reduced instruction set computing* sedangkan CISC adalah *complex instruction set computing*.

AVR dikelompokkan kedalam 4 kelas, yaitu ATtiny, keluarga AT90Sxx, keluarga ATmega, dan keluarga AT86RFxx. Dari kesemua kelas yang membedakan satu sama lain adalah ukuran *on-board* memori, *on-board peripheral* dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan keempat kelas AVR bisa dikatakan hampir sama.

2.6.1 Arsitektur ATmega 8535

Adapun spesifikasi arsitektur yang terdapat pada ATmega 8535 sebagai berikut :

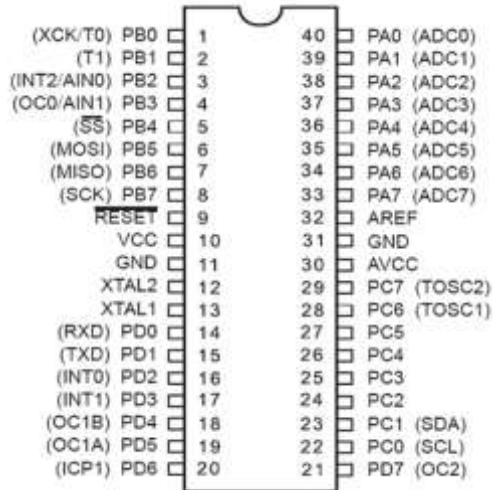
- a. Saluran IO sebanyak 32 buah, yaitu *port A*, *port B*, *port C* dan *port D*
- b. ADC 10 bit sebanyak 8 *channel*
- c. Tiga buah *timer / counter*
- d. 32 *register*
- e. *Watchdog timer* dengan *oscillator* internal
- f. SRAM sebanyak 512 *byte*
- g. Memori *Flash* sebesar 8 kb
- h. Sumber *interrupt* internal dan eksternal
- i. *Port SPI (Serial Peripheral Interface)*
- j. EEPROM on board sebanyak 512 *byte*
- k. Komparator *analog*

- l. *Port USART (Universal Shynchronous Ashynchronous Receiver Transmitter)*

2.6.2 Konfigurasi *Pin* ATMega8535

Mikrokontroler ATMega 8535 memiliki 40 *pin* dan 4 *port* dengan susunan seperti pada gambar 2.10 dengan masing-masing dapat dijelaskan sebagai berikut:

- a. VCC merupakan *pin* yang berfungsi sebagai *pin* masukan catu daya
- b. GND merupakan *pin* *ground*
- c. *Port A* (PA0...PA7) merupakan *pin* I/O dan *pin* masukan ADC
- d. *Port B* (PB0...PB7) merupakan *pin* I/O dan *pin* yang mempunyai fungsi khusus yaitu *timer/counter*, komparator analog dan SPI
- e. *Port C* (PC0...PC7) merupakan *port* I/O dan *pin* yang mempunyai fungsi khusus, yaitu komparator analog dan *timer oscillator*
- f. *Port D* (PD0...PD1) merupakan *port* I/O dan *pin* fungsi khusus yaitu komparator analog dan *interrupt eksternal* serta komunikasi serial
- g. RESET merupakan *pin* yang digunakan untuk mereset mikrokontroler
- h. XTAL1 dan XTAL2 merupakan *pin* masukan *clock eksternal*
- i. AVCC merupakan *pin* masukan untuk tegangan ADC
- j. AREF merupakan *pin* masukan tegangan referensi untuk ADC



Gambar 2.4 Konfigurasi *pin* IC ATmega 8535^[7]

Port A

Pada port A merupakan 8-bit *directional port* I/O. Setiap *pin*nya dapat menyediakan *internal pull-up* resistor (dapat diatur per bit). *Output buffer* Port A dapat memberi arus 20 mA dan dapat mengendalikan *display* LED secara langsung. *Data Direction Register* port A (DDRA) harus *disetting* terlebih dahulu sebelum Port A digunakan. Bit-bit DDRA diisi 0 jika ingin memfungsikan *pin-pin* port A yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, kedelapan *pin* port A juga digunakan untuk masukan sinyal analog bagi A/D *converter*.

Port B

Pada Port B merupakan 8-bit *directional port* I/O. Setiap *pin*nya dapat menyediakan *internal pull-up* resistor (dapat diatur per bit). *Output buffer* Port B

dapat memberi arus 20 mA dan dapat mengendalikan *display* LED secara langsung. Data Direction Register *port* B (DDRB) harus disetting terlebih dahulu sebelum *Port* B digunakan. Bit-bit DDRB diisi 0 jika ingin memfungsikan *pin-pin port* B yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. *Pin-pin port* B juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut 2.1 *Pin-pin port I/O PORT B*.

Table 2.1 *Pin-pin Port I/O PORT B*

<i>Port Pin</i>	Fungsi Khusus
PB0	T0 = <i>timer/counter 0 external counter input</i>
PB1	T1 = <i>timer/counter 0 external counter input</i>
PB2	AIN0 = <i>analog comparator positive input</i>
PB3	AIN1 = <i>analog comparator negative input</i>
PB4	SS = <i>SPI slave select input</i>
PB5	MOSI = <i>SPI bus master output / slave input</i>
PB6	MISO = <i>SPI bus master input / slave output</i>
PB7	SCK = <i>SPI bus serial clock</i>

Port C

Pada *Port C* merupakan 8-bit *directional port I/O*. Setiap *pinnya* dapat menyediakan *internal pull-up* resistor (dapat diatur per bit). *Output buffer Port C* dapat memberi arus 20 mA dan dapat mengendalikan *display* LED secara langsung.

Data *Direction Register port C* (DDRC) harus disetting terlebih dahulu sebelum *Port C* digunakan. Bit-bit

DDRC diisi 0 jika ingin memfungsikan *pin-pin port C* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, dua *pin port C* (PC6 dan PC7) juga memiliki fungsi alternatif sebagai *oscillator* untuk *timer/counter 2*.

Port D

Pada Port D merupakan 8-bit *directional port I/O*. Setiap *pinnya* dapat menyediakan *internal pull-up resistor* (dapat diatur per bit). *Output buffer Port D* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. Data *direction register port D* (DDRD) harus di *setting* terlebih dahulu sebelum *Port D* digunakan. *Pin* pada *port D* juga memiliki untuk fungsi yaitu. ^[7]

Table 2.2 *Pin – pin Port I/O PORT D*

<i>Port</i>	Fungsi khusus
PD0	RDX (UART <i>input</i> line)
PD1	TDX (UART <i>output</i> line)
PD2	INT0 (external interrupt 0 <i>input</i>)
PD3	INT1 (external interrupt 1 <i>input</i>)
PD4	OC1B (Timer <i>output</i> compareB match <i>output</i>)
PD5	OC1A (Counter1 <i>output</i> compareA match <i>output</i>)
PD6	ICP (Timer/counter1 <i>input</i> capture <i>pin</i>)
PD7	OC2 (Timer <i>output</i> compare match <i>output</i>)

2.6.1 Catu Daya

Catu daya merupakan daya untuk menjalankan peralatan elektronik yang dapat diperoleh dari baterai atau sumber lainnya.

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power supply* dapat menggunakan *adaptor* DC atau baterai. *Adaptor* dapat dikoneksikan dengan menggunakan *jack adaptor* pada koneksi *port* input *supply*. *Board* arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 7-12 volt. Jika *supply* kurang dari 7V akan menyebabkan *board* menjadi tidak stabil. Jika menggunakan lebih dari 12V, tegangan di regulator akan sangat panas dan dapat menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 hingga 12 volt. Penjelasan pada pin *power* adalah sebagai berikut :

- **Vin**
Tegangan input ke *board* arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau tegangan suplai menggunakan power jack, aksesnya menggunakan pin ini.
- **5V**
Regulasi *power supply* digunakan untuk power mikrokontroller dan komponen lainnya pada *board*. 5V dapat melalui Vin menggunakan regulator pada *board*, atau *supply* oleh USB atau *supply* regulasi 5V lainnya.
- **3V3**
Suplai 3.3 volt didapat oleh FTDI chip yang ada di *board*. Arus maximumnya adalah 50 Ma.
- **Pin Ground**
Pin ground berfungsi sebagai jalur ground pada arduino.

2.6.2 Memori

Atmega328 memiliki 32 KB *flash* memori untuk menyimpan kode dan 2 KB yang digunakan untuk *bootloader*. Atmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

2.6.3 Input dan Output

Setiap 14 pin digital pada arduino dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maksimum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20-50 Kohms. Adapun beberapa fungsi yang dimiliki pin adalah sebagai berikut :

- Serial : 0 (RX) dan 1 (TX), digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB PTDI ke TTL chip serial.
- Interrupt eksternal : 2 dan 3, pin ini dapat dikonfigurasi untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
- PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi `analogWrite()`.
- SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasan arduino.
- LED : 13, digunakan untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup dan ketika pin LOW, LED mati.

]

2.7 Sistem Pengendalian *On-off*

Sistem pengendalian proses terdiri dari beberapa kerja alat-alat yang digunakan untuk mengendalikan variabel-variabel proses terukur pada suatu nilai *set point* tertentu. Sistem pengendalian proses terdiri dari dua macam, yaitu pengendalian manual dan otomatis. Variabel-variabel proses yang dikendalikan terdiri dari *pressure*, *flow*, *temperature* dan level. Beberapa istilah dalam *system* pengendalian proses antara lain:

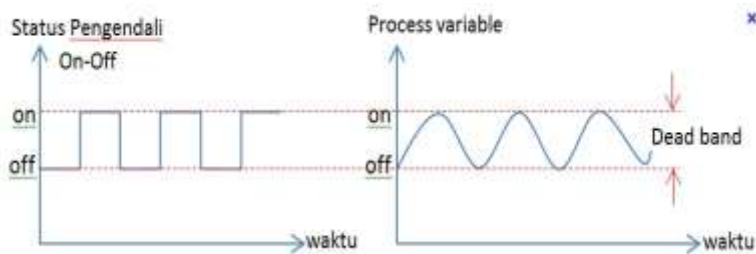
- *Controller* : elemendalam *system* pengendalian yang melakukan pengukuran, perbandingan, perhitungan, dan

koreksi. Salah satu metode *controller* dalam *system* pengendalian proses di industry yaitu *PID (Proportional Integral Derrivatif) control*.

- Proses : Gabungan peralatan yang bekerja sebagai suatu sistem
- Transmitter berfungsi membaca sinyal sensing element dari sensor dan mengubahnya sebelum diteruskan ke *controller*.
- FCE (*Final Control Element*) : instrument akhir dalam *system* pengendalian proses yang berfungsi merubah *variable* pengukuran (*measurement variable*) menjadi *variable* termanipulasi (*manipulated variable*) berdasarkan perintah *controller* terpasang.

Dalam perancangan sebuah pengendalian proses, dibutuhkan suatu penetapan mode proses terlebih dahulu. Hal ini dikarenakan mode pengendalian proses merupakan hal yang paling utama dalam sebuah sistem pengendalian variabel. Salah satu mode pengendalian yang sering juga digunakan dalam banyak pengendalian variabel proses yaitu mode pengendalian *on/off*. Pengendalian *on /off* hanya bekerja pada dua posisi, yaitu posisi “*on*” dan posisi “*off*”. Pada sistem pengendalian *on/off* tidak akan pernah bekerja pada daerah antara 0 hingga 100%. Karena kerjanya yang *on/off* , hasil pengendalian pengendali *on-off* akan menyebabkan proses *variable* yang bergelombang, tidak pernah konstan. Perubahan proses *variable* akan seirama dengan perubahan posisi *final control element*. Besar kecilnya fluktuasi proses *variable* ditentukan oleh titik dimana *controller* “*on*” dan titik dimana “*off*”. Karena karakteristik kerjanya yang hanya *on* dan *off*, *controller* jenis *on-off* juga sering disebut sebagai *two position controller*, *gap controller* atau *snap controller* . Kata *snap* secara harfiah berarti menampar. Sebuah *controller on-off* kemudian juga lazim disebut *snap controller*. Ungkapan kata *snap action* kelak akan juga dipakai untuk kerja *controller* jenis lain yang karena besarnya gain menjadi bekerja secara *on-off*.

Kerja pengendalian *on-off* , seringkali didapatkan dengan memanfaatkan *deab band* suatu proses *switch*. Kerja pengendalian *on-off* banyak dipakai di sistem pengendalian yang sederhana. Namun, tidak semua proses dapat dikendalikan secara *on-off* karena banyak operasi proses yang tidak dapat mentolerir fluktuasi proses *variable*. Jadi, syarat utama untuk memakai pengendali *on-off* yaitu karena proses memang tidak dapat mentolerir fluktuasi proses *variable* pada batas-batas kerja pengendalian *on-off*. Aksi pengendalian dari *controller* ini hanya mempunyai dua kedudukan, maksimum atau minimum, tergantung dari *variable* terkontrolnya, apakah lebih besar atau lebih kecil dari set point.



Gambar 2.5. Fluktuasi Pengendalian *On-Off* ^[15]

2.8 Lampu Indikator

Lampu pijar adalah sumber cahaya buatan yang dihasilkan melalui penyaluran arus listrik melalui *filamen* yang kemudian memanaskan dan menghasilkan cahaya. Kaca yang menyelubungi *filamen* panas tersebut menghalangi udara untuk berhubungan dengannya sehingga *filamen* tidak akan langsung rusak akibat teroksidasi. Lampu pijar dipasarkan dalam berbagai macam bentuk dan tersedia untuk tegangan (*voltase*) kerja yang bervariasi dari mulai 1,25 volt hingga 300 volt. Energi listrik yang diperlukan lampu pijar untuk menghasilkan cahaya yang terang lebih besar dibandingkan dengan sumber cahaya buatan lainnya seperti lampu pijar dan dioda cahaya, maka secara

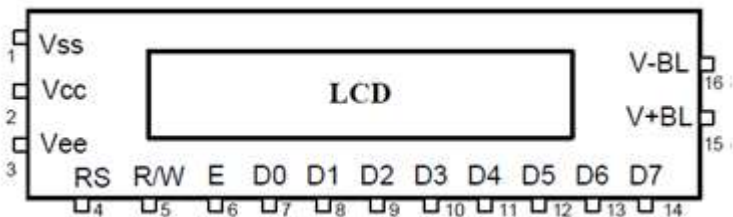
bertahap pada beberapa negara peredaran lampu pijar mulai dibatasi.



Gambar 2.6. Lampu indikator ^[16]

2.9 LCD (*Liquid Cristal Display*)

LCD (*Liquid Cristal Display*) adalah sebuah teknologi yang berfungsi sebagai penampil atau monitor . Modul LCD dengan tampilan 16x2, terdiri dari dua bagian. Pada bagian pertama merupakan panel LCD sebagai media untuk menampilkan informasi berbentuk huruf, simbol maupun angka. LCD 16x2 dapat menampung dua baris, dimana tiap baris dapat menampilkan 16 karakter. Bagian kedua merupakan rangkaian elektronika yang dibentuk dengan mikrokontroler, berada di balik panel LCD. Bagian ini mempunyai fungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi LCD dengan mikrokontroler. Berikut ini konfigurasi LCD 2x16 serta penjelasannya:



Gambar 2.7. Konfigurasi LCD 2x16 ^[17]

Tabel 2.3. Konfigurasi LCD 2x16

Pin	<i>Symbols and Functions</i>
1	GND
2	VCC (+5V)
3	<i>Contrast Adjust</i>
4	(RS) == >> 0 = <i>Instruction input</i> / 1 = <i>Data input</i>
5	(R/W) == >> 0 = <i>Write to LCD Module</i> / 1 = <i>Read from LCD Module</i>
6	(E) == >> <i>Enable Signal</i>
7	(DB0) == >> <i>Data Pin 0</i>
8	(DB1) == >> <i>Data Pin 1</i>
9	(DB2) == >> <i>Data Pin 2</i>
10	(DB3) == >> <i>Data Pin 3</i>
11	(DB4) == >> <i>Data Pin 4</i>
12	(DB5) == >> <i>Data Pin 5</i>
13	(DB6) == >> <i>Data Pin 6</i>

Tabel 2.5. Konfigurasi LCD 2x16 (Lanjutan)

Pin	<i>Symbols and Functions</i>
14	(DB7) == >> <i>Data Pin 7</i>
15	(VB+) == >> <i>Back Light (+5V)</i>

16	(VB-) == >> <i>Back Light</i> (GND)
----	-------------------------------------

Berikut ini karakteristik dari LCD (16x2) :

1. Tampilan 16 karakter, 2 baris.
2. ROM pembangkit karakter 192 jenis.
3. RAM pembangkit karakter 8 jenis (di-program pemakai).
4. RAM data tampilan 80 x 8 bit (8 karakter).
5. *Duty ratio* 1/16.
6. RAM data tampilan dan RAM pembangkit karakter dapat dibaca dari unit mikro-prosesor.
7. Beberapa fungsi perintah antara lain adalah penghapusan tampilan (*display clear*), posisi kursor awal (*crusor home*), tampilan karakter kedip (*display character blink*), penggeseran kursor (*crusor shift*) dan penggeseran tampilan (*display shift*).
8. Rangkaian pembangkit detak (*clock*).
9. Rangkaian otomatis reset saat daya dinyalakan.
10. Catu daya tunggal +5 volt.

2.10 Relay

Relay adalah peralatan yang menggunakan elektromagnet dalam memberikan gaya untuk membuka atau menutup *switch*. Dengan kata lain, suatu *switch* dengan menggunakan tenaga listrik. Suatu *switch* atau *relay* pada saat keadaan tidak fiktif memiliki dua kondisi yaitu NO (*Normally Open*) dan NC (*Normally Close*). Dalam pemilihan suatu *relay* yang harus diperhatikan adalah kapasitas arusnya. *Relay* merupakan piranti control yang dapat berguna untuk menutup dan membuka kontak. *Relay* mekanis digunakan untuk menyambung atau memutuskan beban listrik. Proses *swithing* ini dikontrol oleh rangkaian elektrik. *Relay magnetic* sering digunakan untuk mengontrol *relay* yang lain atau beban dengan daya yang kecil. Seringnya pengulangan membuka dan menutup, kontak dapat mengalami kerusakan akibat dari bunga api dan gesekan mekanis, sehingga

bagian dari kontak tersebut dapat diganti dengan kontak yang baru. Hal ini sering terjadi terutama pada kontak *magnetic*. Material-material kontak yang sering digunakan adalah logam-logam khusus seperti Tembaga (Ag), Emas (Au), Platina (Pt), Nikel (Au-Ni) dan gabungan senyawa-senyawa seperti Ag-Au-Ni.



Gambar 2.8. *Relay DC* ^[18]

Sumber tegangan yang dipakai untuk dialirkan ke coil supaya terjadi gaya elektromagnetik adalah:

- **Sumber arus searah (DC)**

Standar tegangan untuk *relay DC* adalah 6, 12, 24, 48, dan 100 (volt). Kinerja *relay DC* lebih mantap karena kecepatan *switching relay DC* lebih rendah dibandingkan dengan *relay AC* karena induktansi dari coil menekan kecepatan menaikkan arus. Kerugiannya adalah memerlukan catu daya DC yang khusus.

- **Sumber arus bolak balik (AC)**

Relay AC biasanya dieksitasi dengan sumber tegangan 100 atau 200 (V) dengan frekuensi 50 atau 60 (Hz). Pada arus bolak-balik panas dapat terjadi pada kumparan dan inti besi. Untuk catu tegangan yang lebih rendah dari tegangan minimum yang diijinkan akan terjadi desah dan kinerjanya tidak stabil. Untuk sumber daya arus searah (DC) lebih stabil artinya pada coil tidak terjadi getaran karena sumber DC tidak dipengaruhi oleh adanya frekuensi. Pada *relay DC* ini kontaktornya tidak bergetar

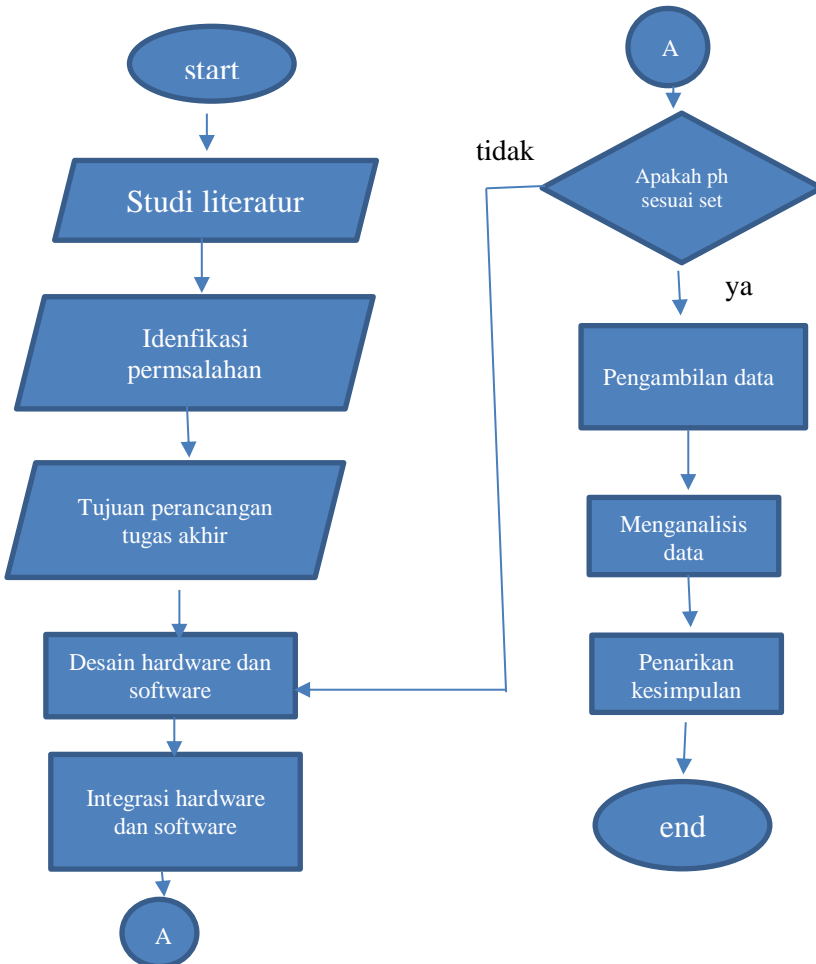
sehingga mempunyai usia pakai yang lama. Untuk sumber daya arus bolak-balik (AC) kurang stabil sehingga terjadi *cattering* atau getaran pada kontaknya karena sumber daya AC pada koil yang mempunyai frekuensi yaitu antara 50 – 60 Hz.

(Halaman ini sengaja dikosongkan)

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Blok Diagram Perancangan Alat

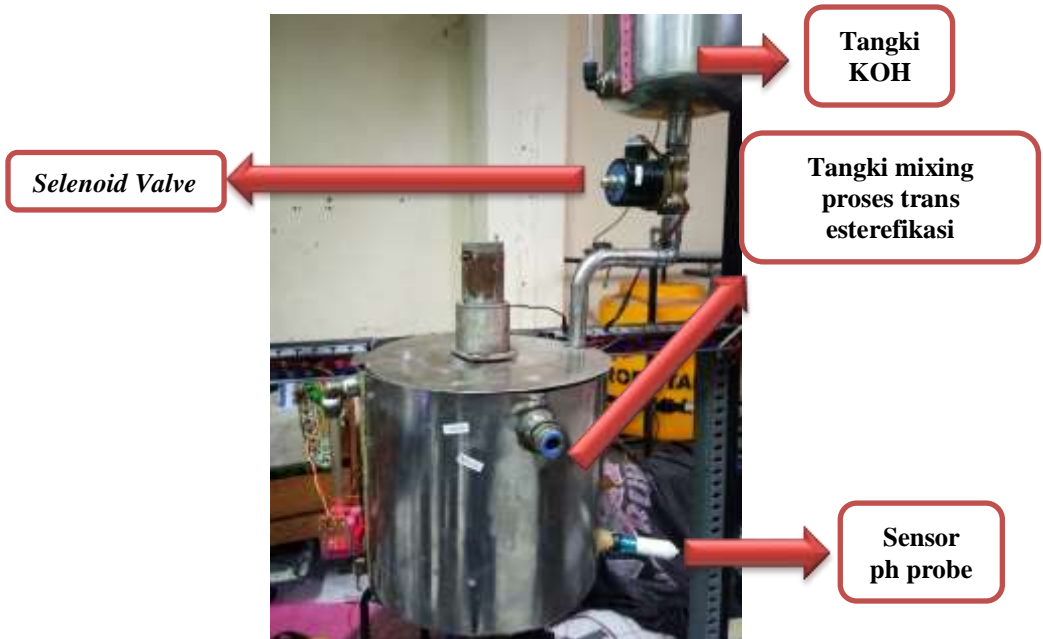
Konsep dasar perancangan dan pembuatan sistem pengendalian *ph* dilakukan dengan tahap-tahap sebagai berikut.



Gambar 3.1. Blok diagram perancangan dan pembuatan alat

3.2 Perancangan Sistem Pengendalian *ph* Pada unit *mixing* Pada Proses trans esterifikasi mini plant biodiesel

Pada perancangan sistem kontrol *plant ph* pada unit *mixing* pada proses trans esterifikasi, mengatur variabel kontrol yaitu *ph*,



Gambar 3.2. Rancangan *Plant ph* Pada Tangki Pencampuran

Pada proses trans esterifikasi mini *plant biodiesel* menggunakan input dari larutan (campuran *methanol* dan *koh*) dan minyak mentah yang nantinya akan masuk ke tangki *mixing*. Pada tangki *mixing* nantinya akan terjadi pencampuran (*methanol* *KOH*) dan minyak mentah yang kemudian akan dideteksi oleh sensor *ph*. Sensor *ph probe* pada *plant* ini terdapat 2 fungsi, yaitu berfungsi untuk memonitoring dan menampilkan *ph* minyak pada tangki pencampuran serta untuk mengontrol kinerja *Solenoid Valve*, ketika *Solenoid Valve* harus berhenti (*off*) dan ketika *Solenoid Valve* harus bekerja (*on*).



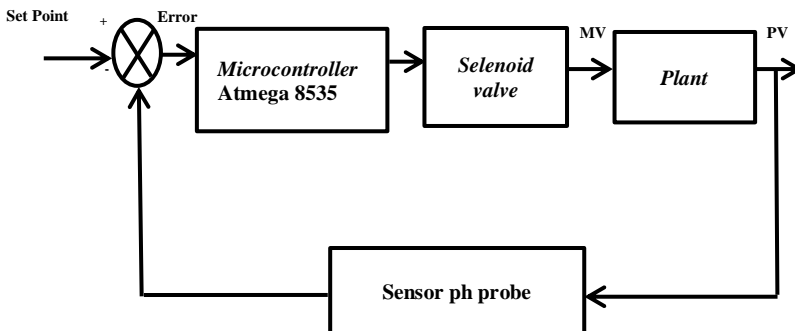
Gambar 3.3. *Solenoid Valve*



Gambar 3.4. *Sensor ph*

Dari sensor ph akan mengirimkan sinyal berupa sinyal analog menuju *microcontroller* Atmega 8535 yang berfungsi sebagai *controller*. Pada *microcontroller* Atmega 8535, data dari sensor akan diolah menjadi sinyal digital (ADC). Kemudian output dari Atmega akan mengaktifkan relay dan mengalirkan output tegangan sesuai dengan tegangan input yang dibutuhkan oleh *solenoid* sehingga *solenoid* dapat menyala. Jika data yang masuk ke *microcontroller* Atmega 8535 lebih dari *set point* (9) maka *microcontroller* akan memerintahkan *solenoid* untuk berhenti (*off*),

begitu juga sebaliknya. Jika data yang masuk ke *microcontroller* kurang dari *set point* (9) maka *microcontroller* akan memerintahkan *solenoid valve* untuk aktif (*on*). Rangkaian yang menyambungkan dari Atmega 8535 menuju *solenoid* yaitu *comparator* dan *modul relay* yang berfungsi untuk mengalirkan tegangan yang dibutuhkan oleh *solenoid* (220VAC) sehingga *solenoid* dapat aktif. Selain itu, peningkatan temperatur yang terjadi pada tangki akan ditampilkan melalui LCD 2 x 16. Sensor RTD diletakkan di Pin Analog Output Atmega 8535, kemudian LCD diletakkan di Pin 1,2, Atmega 8535, serta heater sebagai aktuatur yang diletakkan di pin 7. Berikut merupakan diagram blok pengendalian temperatur.

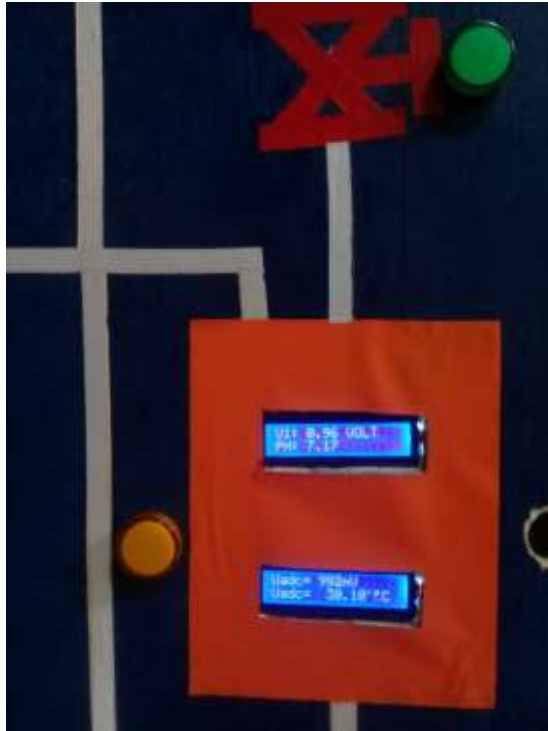


Gambar 3.5. Diagram Blok Pengendalian *ph*

3.3 Perancangan *Local Control Unit* (LCU)

Perancangan LCU ini terdiri dari 2 bagian yaitu perancangan *hardware* dan *software*. Untuk perancangan *hardware* dimulai dari rancangan *plant* pengendalian *ph*, *power supply*, rancangan sensor pada *plant* pengendalian *ph*, rangkaian *microcontroller* Atmega 8535 sebagai *controller*, rangkaian *modul relay* dan *relay*, dan penampilan data pada LCD. Untuk perancangan *software* dimulai dari perancangan *listing code* pada *software* Atmega. Berikut merupakan gambar LCU pada pengendalian *ph* tangki

pencampuran pada proses trans esterifikasi pada *mini plant biodiesel*.



Gambar 3.6. LCU *Plant* Pengendalian *ph*

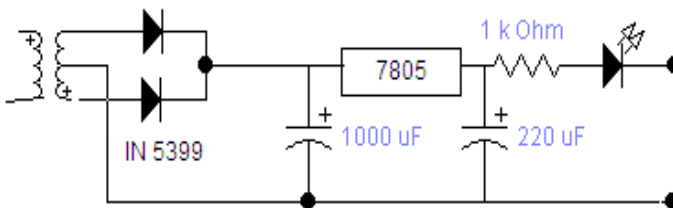
3.4 Perancangan *Hardware*

Dalam pembuatan *plant* pengendalian *ph* pada tangki perlu adanya perancangan *hardware* meliputi *power supply*, sensor, perancangan tangki, *microcontroller*, rangkaian *modul relay*, dan LCD.

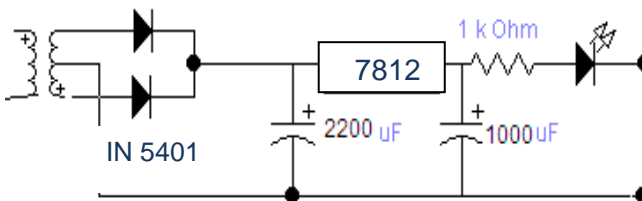
3.4.1 Perancangan *Power Supply*

Dalam pembuatan rangkaian *power supply* terdapat beberapa komponen dalam perangkaian ini yaitu regulator yang mempunyai tegangan IC 7805 dan IC 7812. Regulator IC 7805 dan regulator IC 7812 dapat mengeluarkan tegangan 5 Volt dan 12 Volt DC.

Untuk menghasilkan tegangan 5 Volt terdiri dari beberapa komponen yaitu, dioda IN 5399 yang merupakan dioda yang dapat melewati arus maksimal 2 A, kapasitor 1000 μF dan 220 μF , serta IC 7805. Sedangkan dioda IN 5401 berfungsi untuk menghasilkan tegangan sebesar 12 Volt dengan arus maksimal 3 A. Selain itu, membutuhkan kapasitor 2200 μF dan 1000 μF , serta IC 7824. Dioda yang dipakai dalam rangkaian mempunyai fungsi yang spesifik yaitu untuk menjadikan sinyal AC sinusoidal yang melewatinya menjadi sinyal DC setengah gelombang (*half wave*).



Gambar 3.7. Rangkaian *Power Supply* 5 Volt



Gambar 3.8 Rangkaian *Power Supply* 12 Volt

Kapasitor mempunyai fungsi untuk memperhalus sinyal DC dari dioda. Kemudian sinyal DC keluaran dari kapasitor akan dimasukkan ke IC 7805, 7812. Hasil keluaran dari IC 7805 dan 7812 adalah tegangan 5 Volt dan 12 Volt.

3.4.2 Perancangan Sensor ph probe

Sensor ph bekerja dengan cara sensor akan melakukan penginderaan pada saat perubahan ph setiap perubahan senyaawa akan menunjukkan tegangan sebesar 10 mV. Prinsip kerja sensor ph

pada *plant* ini adalah ketika *ph* yang dideteksi oleh sensor maka sensor *ph* akan mengirimkan sinyal berupa sinyal *analog* menuju *microcontroller* Atmega 8535 yang berfungsi sebagai *controller*. Pada *microcontroller* Atmega 8535, data dari sensor akan diolah menjadi sinyal *digital* (ADC). Kemudian output dari akan mengaktifkan *relay* dan mengalirkan output tegangan sesuai dengan tegangan input yang dibutuhkan oleh *solenoid* sehingga *solenoid* dapat menyala. Untuk memperjelas rangkaian sensor *ph* dapat dilihat gambar dibawah ini.



Gambar 3.9. Penempatan Sensor *ph*

Sensor *ph* berfungsi sebagai pendeteksi *ph* tangki pencampuran (*mixing*) seperti pada gambar di atas dan ditampilkan pada LCD 2 X 16, sehingga *user* dapat mengetahui peningkatan dan penurunan *ph* setiap detik atau menit. Selain itu, Sensor *ph* berfungsi untuk mengontrol *on/off solenoid valve*. Output dari Sensor *ph* akan masuk ke *microcontroller* sebagai input.

3.4.3 Modul relay

Fungsi dari *modul relay* adalah untuk konversi dari output keluaran arduino uno menuju aktuator. Keluaran dari arduino adalah 5V DC sedangkan yang dibutuhkan *solenoid valve* adalah tegangan 220 AC. Maka dari itu dibutuhkan *modul relay* untuk mengkonversi dari 5V DC menjadi 220 AC. Cara kerjanya adalah dengan memberikan VCC (5V DC) dan *ground* untuk menyalakan *modul relay*. Jika sudah yang dibutuhkan untuk menggerakkan *modul relay* adalah 5V DC agar *relay* dapat berpindah dari NC menjadi NO.



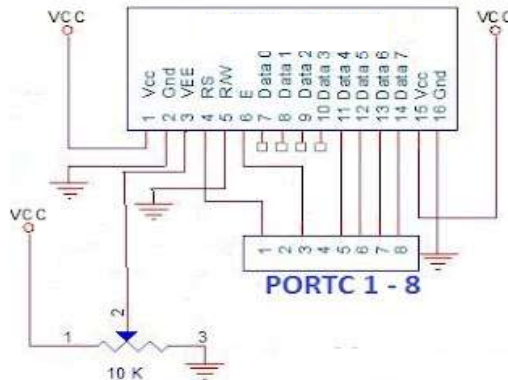
Gambar 3.10. Rangkaian *Modul Relay*

3.4.4 Perancangan *Display LCD (Liquid Crystal Display)*

LCD yang digunakan 2 baris x 16 kolom. LCD memiliki memori internal yang berisi definisi karakter sesuai dengan standar ASCII (CGROM – *Character Generator ROM*) dan memori sementara (RAM) yang bisa digunakan bila memerlukan karakter khusus (berkapasitas 8 karakter). RAM ini juga berfungsi untuk menyimpan karakter yang ingin ditampilkan di LCD.

Pin untuk LCD sendiri adalah pin pengisian data agar data dapat diterima dan diolah melalui *Microcontroller* ke LCD. Sebelum data dibaca oleh LCD maka data diisi oleh RS. Untuk pemrograman mikrokontroler menggunakan software atmel ISP yang merupakan *software* untuk *download* ke mikrokontroler.

Program yang kita tulis dalam *notepad* harus disimpan dalam ekstensi .asm dan kemudian dikonversikan dalam bentuk bilangan heksadesimal yang secara otomatis akan mengoreksi error program. Bila *error* adalah nol maka program bisa di *download* menuju mikrokontroler.



Gambar 3.11. Koneksi LCD 2 X 16 ke *Microcontroller*

Berikut merupakan tampilan perancangan LCD pada uji Sensor pH dimana pada rangkaian tersebut variabel temperatur yang ditampilkan.



Gambar 3.12. Tampilan Perancangan LCD Pada Uji Coba Sensor ph probe df robot

3.4.5 Perancangan *selenoid valve*

Dalam pembuatan plant pengendalian ph dibutuhkan aktuator untuk mengendalikan katalis agar ph dalam tanki dapat sesuai dengan set point, aktuator yang digunakan dalam pengendalian Ph adalah *selenoid valve*. Selenoid valve ada beerapa macam yaitu solenoid yang aktif dengan presure tertentu dan selenoid yang langsung aktif tanpa harus memenuhi presure yang ditetapkan atau memakai selenoid valve yang menggunakan zero perssure pada plant pengendalian pH ini menggunakan selenoid zero *pressure* kaena katalis yang digunakan tidak mempunyai *pressure*, untuk dapat mengetahui *selenoid valve* dengan zero pressure atau tidak dapat dilihat dari *tag name selenoid valve*. Berikut gambar untuk selenoid dengan zero pressure dan tidak :



Gambar 3.13. *selenoid zero pressure*



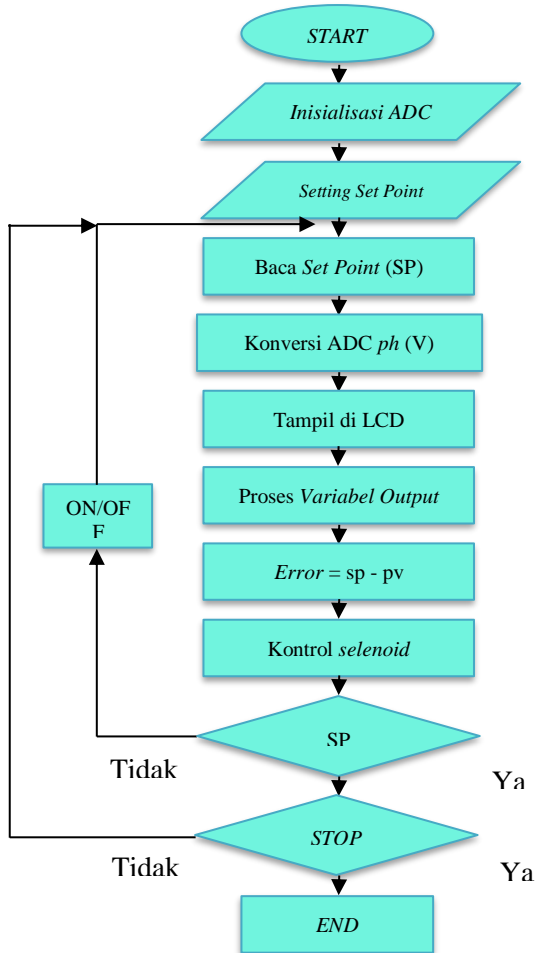
Gambar 3.14. *solenoid dengan pressure*

3.5 Perancangan Perangkat Lunak *Software*

Perancangan *software* yang digunakan pada tugas akhir ini yaitu *microcontroller* Atmega 8535 yang dipakai dan disesuaikan dengan kondisi *hardware* yang digunakan. *Software* yang dipakai terdiri dari beberapa bagian pokok, yaitu :

1. *Software* yang dipakai untuk pembacaan *microcontroller*.
2. *Software* yang digunakan untuk pembacaan output dari sensor oleh *microcontroller* yang digunakan sebagai pengolah data.
3. *Software* penampil ke LCD.

Perancangan *software* digunakan untuk mendukung kerja dari perangkat keras (*hardware*). Pada proses pembuatan listing program menggunakan *Software* Atmega.



Gambar 3.15. Diagram Alir Program

Diagram Alir diatas menjelaskan tentang perancangan *software* pada *microcontroller* Atmega 8535. Input dari ADC yang terdapat pada pin *analog read* yaitu dari *output* Sensor *ph*. Kemudian dalam listing pemrograman dimasukkan *set point*

temperatur yang telah ditentukan. Pin *analog read* yang mempunyai fasilitas ADC akan membaca dan mengkonversi menjadi temperatur pada tangki sesuai *set point* yaitu *set point* dengan *ph* antara 9-10 yang kemudian akan ditampilkan melalui LCD 2 x 16. Dengan adanya pengukuran *ph* pada tangki melalui Sensor *ph* dan *microcontroller* Atmega 8535, data pengukuran yang dihasilkan akan terdapat *error* kemudian dibandingkan dengan *ph* yang sesuai dengan *set point*. Kemudian *microcontroller* Atmega 8535 sebagai *controller* akan bekerja dan mengendalikan *heater* dengan mode *on-off* berdasarkan *set point* tertentu.

3.6 Prosedur Operasional

Pada pengendalian *pH* dalam proses transesterifikasi dibutuhkan tiga komponen penting, yaitu *controller* yang berupa Atmega 8535, *ph probe* sebagai sensor, dan *solenoid* sebagai aktuator. Pengendalian *ph* akan berjalan sesuai dengan standar apabila proses pencampuran larutan minyak mentah dengan larutan campuran (*methanol* dan KOH) pada tangki *mixing* dapat berlangsung selama 15 menit sehingga temperatur harus terjaga dengan baik sesuai *set point*. Untuk mengendalikan temperatur agar selalu berada dalam *set point* maka dibutuhkan Sensor *ph probe* yang terhubung pada Atmega 8535 sebagai *controller* untuk memberikan perintah dan mengendalikan aktuator dengan sistem *on/off*.

Apabila temperatur berada pada keadaan $>9-10$ maka *controller* akan memberikan perintah untuk menghentikan *solenoid* dan sebaliknya, jika temperatur $<9-10$ maka *controller* akan memberikan perintah untuk menyalakan *solenoid* kembali. *ph* yang terjaga sesuai *set point* akan mempengaruhi pada konversi dari reaksi kimia pelarutan minyak biji nyamplung yang telah bereaksi dengan larutan (*methanol* dan KOH) menjadi metil ester (biodiesel) yang dihasilkan. Setelah proses degumming yang berlangsung selama 15 menit maka Atmega 8535 sebagai *controller* akan memberikan perintah pada *solenoid valve* untuk

aktif sehingga dapat membuka menuju proses selanjutnya yaitu proses pengendapan.

BAB IV

PENGUJIAN DAN ANALISA DATA

4.1 Pengujian Alat

Pada bab ini menjelaskan tentang pengujian alat pada tangki pencampuran. Setelah dilakukan perancangan dan pembuatan sistem pengendalian pH pada tangki pencampuran maka perlu dilakukan pengujian terhadap *hardware* dan *software* yang telah dibuat. Selanjutnya dilakukan analisa secara menyeluruh terhadap hasil pengujian. Pengujian ini perlu dilakukan untuk mengetahui performansi alat secara keseluruhan yaitu pada perangkat keras (*hardware*) dan perangkat lunak (*software*).

Sistem kerja pada pengendalian pH tangki pencampuran *mini plant biodiesel process* bertujuan untuk menjaga temperatur dalam tangki pencampuran sesuai dengan *set point*. Pada pengendalian temperatur ini menggunakan sensor PH probe (*ph sensor*) untuk mendeteksi pH ruangan pada tangki pencampuran. Keluaran pH probe berupa tegangan yang kemudian masuk ke ADC pada kaki pin mikrokontroler sebagai kontroler dari sistem pengendalian temperatur. Mikrokontroler inilah yang memberikan perintah *on/off* ke aktuator melalui *relay* yang telah terhubung ke *solenoid valve*. Ketika temperatur dalam ruangan melebihi *set point* pH 9 maka kontroler akan mematikan *solenoid valve*. Sebaliknya ketika pH kurang dari *set point* maka kontroler akan mengaktifkan *solenoid valve*.

4.2 Pengujian Sistem

Pada pengujian sistem ini dilakukan dengan cara memberikan sinyal *input* yang berupa *Proses Variable (PV)* yaitu pH yang berasal dari tangki KOH. Pengujian sistem menggunakan *set point* yang telah ditentukan untuk mengetahui lama waktu proses pencampuran larutan minyak dan larutan metoksida (*methanol* dan KOH). Berikut adalah pengujian sistem pada unit pencampuran

dengan pengukuran perubahan ph yang terjadi dimulai dari PH normal sampai menuju *set point* sebesar ph 9.

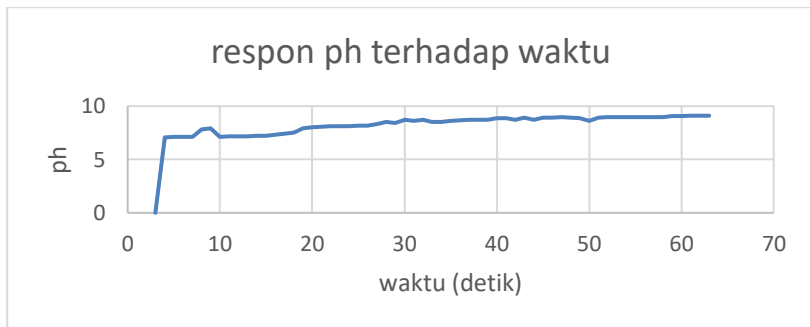
Tabel 4.1. Pengukuran ph Pada *Set point* ph 9

NO	<i>Set point</i> ph	<i>ph</i> 9 (basa)	Waktu (detik)
1	9	7,09	5
2	9	7,1	10
3	9	7,12	15
4	9	7,1	20
5	9	7,8	25
6	9	7,9	30
7	9	7,13	35
8	9	7,15	40
9	9	7,18	45
10	9	7,17	50
11	9	7,20	55
12	9	7,21	60
13	9	7,3	65
14	9	7,4	70
15	9	7,5	75
16	9	7,9	80
17	9	8,03	85
18	9	8,05	90
19	9	8,1	95
20	9	8,13	100
21	9	8,14	105
22	9	8,16	110
23	9	8,19	115
24	9	8,30	120
25	9	8,50	125
26	9	8,4	130
27	9	8,7	135
28	9	8,6	140
29	9	8,7	145

Tabel 4.1. Pengukuran Ph Pada *Set point* ph 9 (Lanjutan)

NO	<i>Set point</i> Ph 9(basa)	<i>ph</i> (basa)	Waktu (detik)
30	9	8,5	150
31	9	8,53	155
32	9	8.6	160
33	9	8,66	165
34	9	8,7	170
35	9	8,73	175
36	9	8,70	180
37	9	8,85	186
38	9	8,88	190
39	9	8,7	195
40	9	8,9	200
41	9	8,7	205
42	9	8,92	210
43	9	8,94	215
44	9	8,97	220
45	9	8,90	225
46	9	8,89	230
47	9	8,64	235
48	9	8,90	240
49	9	8,98	245
50	9	8,98	250
51	9	8,98	255
52	9	8,98	260
53	9	8,98	265
54	9	8,98	270
55	9	8,98	275
56	9	9,07	280
57	9	9,06	285
58	9	9,1	290
59	9	9,1	295
60	9	9,1	300

Dari tabel diatas telah dilakukan pengujian pada plant tangki pencampuran dengan memberikan *set point* dengan ph yang telah ditentukan yaitu ph 9. Dengan cara mengukur perubahan ph yang terdapat pada tangki pencampuran dengan menggunakan sensor ph probe sebagai pendeteksi ph dalam cairan tangki pencampuran larutan minyak dan campuran metoxida (KOH dan *methanol*) dan telah menghasilkan waktu kurang lebih 1 menit. Dengan begitu akan mudah untuk mengatur aktuator *solenoid valve*. Apabila ph berada pada keadaan >9 akan menghentikan *solenoid valve* dan jika $ph < 9$ akan menyalakan *solenoid valve* kembali. Proses transesterifikasi berlangsung selama 1 jam sehingga ph harus berada dalam *set point* dan setelah 1 jam *solenoid valve* akan aktif untuk membuka menuju proses selanjutnya yaitu proses separasi.



Gambar 4.1. Grafik Perubahan ph Terhadap Waktu Pada *set point* ph 9.

4.2.1 Pengujian Sistem Pada *Set point* ph 8

Pada pengujian dibawah ini adalah pengujian plant tangki pencampuran dengan *set point* ph 8 . Percobaan dilakukan dengan mengamati perubahan ph selama 3 menit hingga menghasilkan campuran larutan apakah tercampur secara optimal. Data hasil pengujian dengan *set point* ph 8 adalah sebagai berikut:

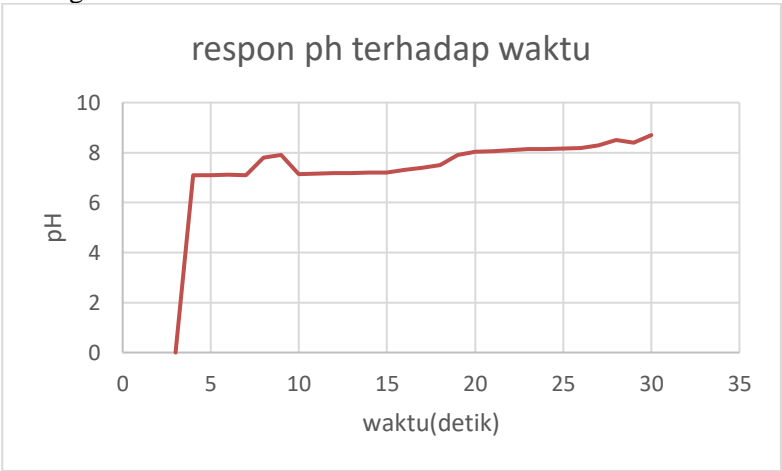
Tabel 4.2. Data Pengujian Sistem dengan *Set point ph 8*

NO	<i>Set point</i>	<i>ph</i> Pembacaan	Waktu (detik)
1	8	7,09	5
2	8	7,1	10
3	8	7,12	15
4	8	7,1	20
5	8	7,8	25
6	8	7,9	30
7	8	7,13	35
8	8	7,15	40
9	8	7,18	45
10	8	7,17	50
11	8	7,20	55
12	8	7,21	60
13	8	7,3	65
14	8	7,4	70
15	8	7,5	75
16	8	7,9	80
17	8	8,03	85
18	8	8,05	90
19	8	8,1	95
20	8	8,13	100
21	8	8,14	105
22	8	8,16	110
23	8	8,19	115
24	8	8,30	120
25	8	8,50	125
26	8	8,4	130
27	8	8,7	135

Tabel 4.2. Data Pengujian Sistem dengan *Set point* ph 8(Lanjutan)

NO	<i>Set point</i>	<i>Ph</i> Pembacaan	Waktu (menit)
28	8	8,5	140
29	8	8,53	145
30	8	8.6	150
31	8	8,66	155
32	8	8,7	160

Pada data di atas merupakan uji sistem dengan *set point* ph 8. Ph awal sebesar 7,09 sampai dengan 8,7. Pada tabel diatas diketahui pula *ts (setling time)* pada sistem ini membutuhkan waktu 1 menit 30 detik untuk mencapai *set point*. Untuk menghasilkan larutan minyak dengan metoxida yang tercampur selama beberapa menit dengan *set point* ph 8 dibutuhkan 1 menit 30 detik dengan pengujian per 5 detik. Grafik pengujian sistem terhadap waktu adalah sebagai berikut :



Gambar 4.2. Grafik Pengujian Sistem Pada *Set point* ph 8

Pada grafik diatas dapat diketahui data *ess* (*error steady state*) yaitu (*Output – set point*) sebagai berikut :

Tabel 4.3. Tabel Data Error (*ess*) dengan *Set point* *ph* 8

Data ke-	<i>Set point</i>	<i>Ph</i> Pembacaan	<i>Ess</i> (Pembacaan <i>Set point</i>)
9	8	8,03	-0,03
10	8	8,05	-0,05
11	8	8,1	-0,1
12	8	8,13	-0,13
13	8	8,14	-0,14
14	8	8,16	-0,16
15	8	8,19	-0,19
16	8	8,30	-0,3
17	8	8,50	-0,5
18	8	8,4	-0,4
19	8	8,7	-0,7
20	8	8,03	-0,03
21	8	8,05	-0,05
22	8	8,1	-0,1
23	8	8,13	-0,13
24	8	8,14	-0,14
25	8	8,16	-0,16
26	8	8,19	-0,19
27	8	8,30	-0,3
28	8	8,50	-0,5
29	8	8,4	-0,4
30	8	8,7	-0,7
31	8	8,03	-0,03
32	8	8,05	-0,05
Rata-rata <i>ess</i>			-0,22833

4.2.2 Pengujian Sistem Dengan *Set point* ph 9

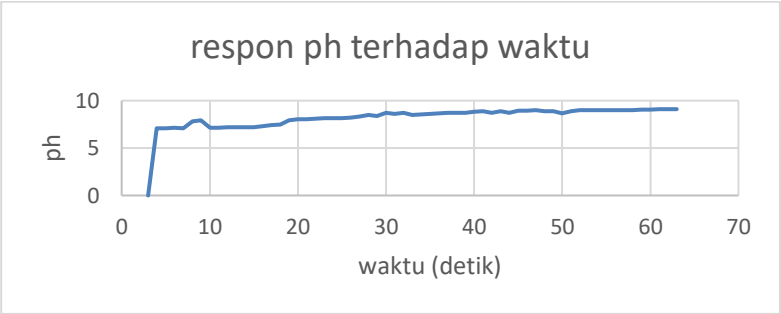
Pada Pengujian dibawah ini adalah pengujian tangki pencampuran dengan *set point* ph 9. Percobaan dilakukan dengan mengamati perubahan temperatur tiap 5 detik sehingga menghasilkan pencampuran larutan dengan baik. Data hasil pengujian dengan *set point* ph 9 adalah sebagai berikut :

Tabel 4.4. Data Pengujian Sistem dengan *Set point* ph 9

NO	<i>Set point</i> ph	ph 9 (basa)	Waktu (detik)
1	9	7,09	5
2	9	7,1	10
3	9	7,12	15
4	9	7,1	20
5	9	7,8	25
6	9	7,9	30
7	9	7,13	35
8	9	7,15	40
9	9	7,18	45
10	9	7,17	50
11	9	7,20	55
12	9	7,21	60
13	9	7,3	65
14	9	7,4	70
15	9	7,5	75
16	9	7,9	80
17	9	8,03	85
18	9	8,05	90
19	9	8,1	95
20	9	8,13	100
21	9	8,14	105
22	9	8,16	110
23	9	8,19	115
24	9	8,30	120
25	9	8,50	125

26	9	8,4	130
27	9	8,7	135
28	9	8,6	140
29	9	8,7	145
30	9	8,5	150
31	9	8,53	155
32	9	8,6	160
33	9	8,66	165
34	9	8,7	170
35	9	8,73	175
36	9	8,70	180
37	9	8,85	186
38	9	8,88	190
39	9	8,7	195
40	9	8,9	200
41	9	8,7	205
42	9	8,92	210
43	9	8,94	215
44	9	8,97	220
45	9	8,90	225
46	9	8,89	230
47	9	8,64	235
48	9	8,90	240
49	9	8,98	245
50	9	8,98	250
51	9	8,98	255
52	9	8,98	260
53	9	8,98	265
54	9	8,98	270
55	9	8,98	275
56	9	9,07	280
57	9	9,06	285
58	9	9,1	290
59	9	9,1	295
60	9	9,1	300

Pada data di atas merupakan uji sistem dengan *set point* 60°C. Temperatur awal sebesar 30°C sampai dengan 60 °C. Pada tabel diatas diketahui pula *ts (setling time)* pada sistem membutuhkan 10 menit untuk mencapai *set point*. Untuk menghasilkan larutan yang tercampur dengan *set point* 60°C dibutuhkan 29 menit dengan pengambilan data sebanyak 30 kali. Grafik pengujian sistem terhadap waktu adalah sebagai berikut :



Gambar 4.3 Grafik Pengujian Sistem Pada *Set point* ph 9

Pada grafik diatas dapat diketahui data *ess (error steady state)* yaitu (*Output – set point*) sebagai berikut :

Tabel 4.5. Tabel Data Error (*ess*) dengan *Set point* ph 9

Data ke-	Set point	ph Pembacaan	Ess (Pembacaan Set point)
56	9	9,07	-0,07
57	9	9,06	-0,06
58	9	9,1	-0,1
59	9	9,1	-0,1
60	9	9,1	-0,1
Rata-rata <i>ess</i>			-0,086

4.2.3 Perhitungan Akurasi

Akurasi yaitu keterdekatan hasil pengukuran alat ukur terhadap harga sebenarnya. Perhitungan akurasi menggunakan rumus sebagai berikut :

$$A = 1 - \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\% \quad (4.1)$$

dengan :

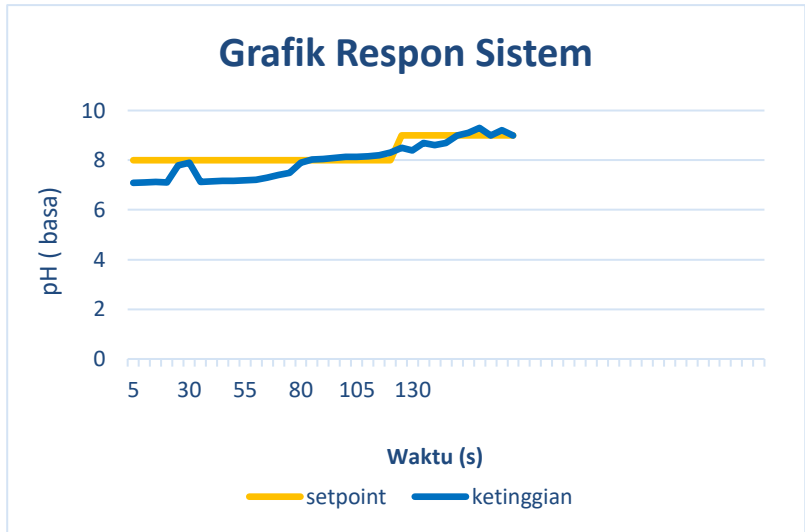
A = Akurasi
 Y_n = Nilai Sebenarnya
 X_n = Nilai Pengukuran

Tabel 4.6. Data perhitungan akurasi

No.	Hasil Pengukuran		Perhitungan Akurasi
	Pembacaan Ph	Pembacaan Ph	
1	3	3,3	93,94 %
2	5	5,3	97,22 %
3	7	7,2	95,00 %
4	9	9,1	95,24 %
5	12	12	100 %
6	13	13	100 %
7	14	14	96,92 %
Rata-rata akurasi relatif			96,90 %

4.2.4 Respon pengendalian pH

Pada sistem pengendalian pH ini digunakan sensor pH probe dan setelah pengujian sensor, serta menghitung nilai ketidakpastian diperluas. Maka dari itu, perlu dicari grafik respon dari sistem pengendalian pH berdasarkan dari data pembacaan sensor.

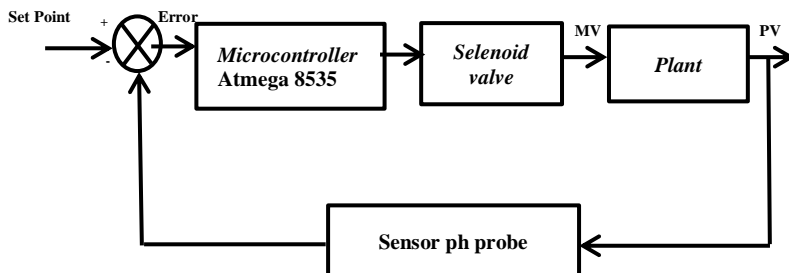


Gambar 4.4 Grafik Respon Sistem Pengendalian pH

Pada **gambar 4.5** merupakan grafik respon sistem antara *set point* level dengan waktu yang dibutuhkan untuk mencapai *set point* tersebut.

Nilai *maximum overshoot* (Mp) dari sistem untuk semua set point adalah 0,3.

4.3 Analisa Data



Gambar 4.5. Diagram Blok Pengendalian *ph*

Pada rancang bangun sistem pengendalian *ph* dalam proses transesterifikasi pada *mini plant* biodiesel proses ini pengujian sistem menggunakan *set point* yang berbeda-beda untuk mengetahui waktu yang dibutuhkan dalam proses pencampuran antara larutan minyak biji kapuk dan metoksida (*methanol* dan *KOH*) yaitu dengan *set point ph* agar menjaga agar larutan basa dengan menggunakan *set point* 8 dan 9. Pengambilan data dilakukan dengan mengamati perubahan *ph* setiap 5 detik sekali.

Pada pengujian sistem dengan *set point* 9 dibutuhkan waktu 3 menit untuk mencapai *set point*. Untuk menghasilkan larutan pencampuran dengan menjaga *ph* ideal dengan *set point ph* 9 dibutuhkan waktu selama 3 menit. Selain itu, pada pengujian tersebut telah didapatkan nilai rata-rata *ess (error steady state)* yaitu (*Output – Set point*) -0,866.

Pada pengujian sistem dengan *set point* 8 dibutuhkan waktu selama 1 menit 30 detik untuk mencapai *set point* dan proses pencampuran larutan. Selain itu, dapat pula ditentukan rata-rata *ess (error steady state)* yaitu (*Output – Set point*) -0,2883.

Berdasarkan data hasil pengujian sistem yang dilakukan menunjukkan bahwa variabel *ph* pada proses transesterifikasi merupakan faktor yang sangat penting untuk dikontrol sehingga *ph* harus berada dalam *set point*. Seperti pada tabel 4.3 pengujian sistem yang dilakukan menggunakan *set point* 8 tidak dapat menghasilkan reaksi larutan pencampuran minyak dengan metoksida dengan baik karena pada *ph* kurang dari 8 konversi menjadi biodiesel yang dihasilkan tidak dapat berjalan maksimal sehingga dapat berpengaruh terhadap produk biodiesel yang dihasilkan. Selain itu, dari pengujian alat yang dilakukan pada *set point ph* 8 terdapat komponen-komponen yang tidak dapat bekerja dengan maksimal dikarenakan pada *ph* diatas *methanol* tidak dapat bereaksi dengan baik. Oleh karena itu untuk *set point ph* yang baik maka dipilih lah *ph* 9 sebagai *set point* untuk pengendalian *ph* agar reaksi yang terjadi dapat terjadi dengan baik tanpa mengurangi produk dari biodiesel. untuk perhitungan akurasi sistem didapatkan bahwa sistem mempunyai akurasi sebesar 96,90%.

Kendala yang didapatkan dalam pengujian data adalah pengambilan data *ph* secara manual dan otomatis, karena pada saat pengambilan data secara manual dan otomatis terdapat perbedaan dari data dan uji sensor sehingga diperlukan pengujian data dengan sangat teliti.

BAB V

PENUTUP

5.1 Kesimpulan

Pada Tugas Akhir ini dapat disimpulkan bahwa:

1. Telah dirancang alat pengendalian PH dengan menggunakan sensor PH probe dfrobot, mikrokontroler Aatmega 8535 sebagai kontroler serta *solenoid valve* sebagai aktuator.
2. Pada hasil percobaan dengan *set point* 8 diperoleh *settling time (ts)* selama 1 menit 30 detik dan rata-rata *error steady state (ess)* sebesar -0,22 serta untuk menghasilkan larutan pencampuran minyak dan metoxida
3. Pada hasil percobaan dengan *set point* 9 diperoleh *settling time (ts)* selama 3 menit dan rata-rata *error steady state (ess)* sebesar -0,086 serta untuk menghasilkan larutan pencampuran minyak .
4. ph optimum yang digunakan untuk proses transesterifikasi adalah pada *set point* 9 karena dapat menghasilkan pencampuran larutan metoxida dan minyak dengan baik. Sistem pengendalian temperatur larutan pencampuran minyak dan metoxida (KOH dan *methanol*) mempunyai akurasi sebesar 96,90 %.

5.2 Saran

Pada tugas akhir sistem pengendalian temperature pada larutan pencampuran minyak dan metoxida (KOH dan *methanol*) dapat ditambahkan *SIS (Safety Instrumented System)* untuk mengetahui bahwa larutan dapat menghasilkan metil ester dan *glycerin* yang maksimal dalam pencampuran serta memberikan pengaman agar temperatur dapat terjaga dengan waktu yang ditentukan.

DAFTAR PUSTAKA

- [1] Suryandari, Ade Sonya., Siska Norma Prasasti & Achmad Roesyadi. 2013. “Pembuatan Biodiesel dari Minyak Biji nyamplung (*Ceiba Pentandra*) Melalui Proses Transesterifikasi dengan Katalis MgO/CaO ”. **Jurnal Teknik POMITS** Volume 2, No 1, Institut Teknologi Sepuluh Nopember (ITS). Surabaya.
- [2] Sofyan Mudzofar., Ilham Tanjung & Herry Santosa. 2014. “Optimasi Variabel Yang Paling Berpengaruh Pada Pembuatan Biodiesel Dari Minyak Biji Randu Dengan Proses Transesterifikasi”. **Jurnal Teknik**, Universitas Diponegoro. Semarang.
- [3] Tanpa Nama. Tanpa Tahun. **Bab I Pendahuluan**, <<http://digilib.its.ac.id/public/ITS-Undergraduate-37232-2862300333-Chapter1.pdf>>.
- [4] Frederic, Poltak Nixon. 2013. “Pembuatan Biodiesel Dari Minyak Biji Kapok Dengan Proses Esterifikasi Transesterifikasi”. **Jurnal Teknologi Kimia dan Industri** Volume 2, No. 2, Universitas Diponegoro. Semarang.
- [5] Tanpa Nama. Tanpa Tahun. **Bab II Tinjauan Pustaka**, <<http://repository.usu.ac.id/bitstream/123456789/35009/4/Chapter%20II.pdf>>.
- [6] Tanpa Nama. Tanpa Tahun. **Bab II Tinjauan Pustaka**, <eprints.undip.ac.id/36596/4/Bab_II,_Biodisel.pdf>
- [7] Ghozali Lutfi Eko., 2013. “Rancang Bangun Sistem Level Dan Pressure Steam Generator Pada Simulator Mixing Process di Workshop Instrumentasi”. **Jurnal Teknik POMITS** Volume 2, No 1, Institut Teknologi Sepuluh Nopember (ITS). Surabaya.

- [8] Tanpa Nama. Tanpa Tahun. **Biodiesel Dari Minyak Nabati**,
<<http://che.unsyiah.ac.id/wpcontent/uploads/sites/4/2014/01/Biodiesel-Dari-Minyak-Nabati.pdf>>.
- [9] <http://telinks.wordpress.com/2010/04/09/rangkaian-sensor-suhu-lm35/> [ONLINE]
- [10] Tanpa Nama. Tanpa Tahun. **Ph probe Precision Centigrade ph Sensors**,
< <http://www.ti.com/lit/ds/symlink/lm35.pdf>>.
- [11] <http://depokinstruments.com/2010/08/09/produk-baru-di-wlm35ts-di-waterproof-lm35-temperature-sensor/> [ONLINE]
- [12] <Http://id.prmob.net/solenoida/katup/solenoid-valve-917220.html> [ONLINE]
- [13] Tanpa Nama. Tanpa Tahun. **Bab II Landasan Teori**,
<<http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2011-2-01650-SK%20Bab2001.pdf>>.
- [14] Malvino., 1996. “Prinsip-Prinsip Elektroniks”. Jakarta, Erlangga.
- [15] Gunterus, Frans., 1997. “Falsafah Dasar Sistem Pengendalian Proses”. Elex Media Komputindo, Jakarta, 1997.
- [16] https://id.wikipedia.org/wiki/Lampu_pijar [ONLINE]
- [17] Susanto,Dimas., 2009. “Pemrograman Mikrokontroller AVR ATmega8535 dan LCD Board 2x16”. Bandung.
- [18] Wicaksono,Handy., Tanpa Tahun. “*Relay*-Prinsip dan Aplikasi” , Teknik Elektro. Universitas Kristen Petra.

DATA SHEET ATMEGA 8535

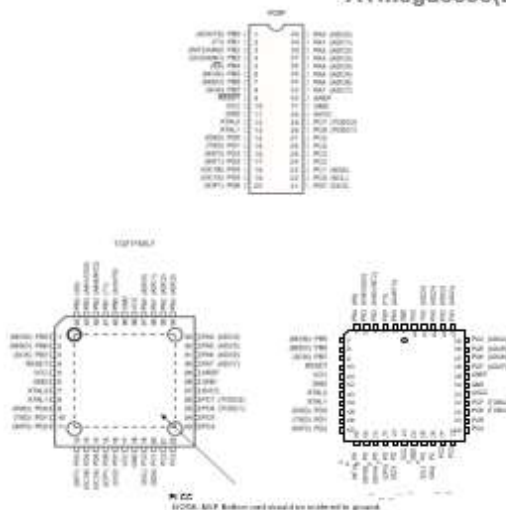


- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 139 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPs Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memory
 - 8K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 512 Bytes Internal SRAM
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels for IQ1* Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 20x for IQ1* Package Only
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin TQFP, 44-lead PQFP, 44-lead LQFP, and 44-pin QFNMLP
- Operating Voltages
 - 2.7 - 5.5V for ATmega8535L
 - 4.5 - 5.5V for ATmega8535
- Speed Grades
 - 0 - 8 MHz for ATmega8535L
 - 0 - 16 MHz for ATmega8535

Pin Configurations

Figure 1. Pinout ATmega8535

ATmega8535(L)



Disclaimer

Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

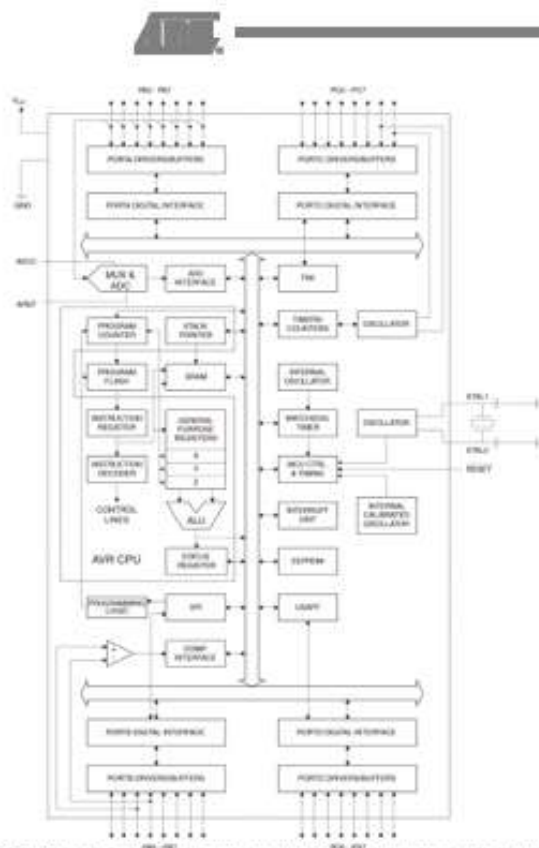
Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible



The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain in TQFP package, a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the asynchronous timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

AT90S8535 Compatibility

The ATmega8535 provides all the features of the AT90S8535. In addition, several new features are added. The ATmega8535 is backward compatible with AT90S8535 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S8535 compatibility mode can be selected by programming the S8535C fuse. ATmega8535 is pin compatible with AT90S8535, and can replace the AT90S8535 on current Printed Circuit Boards. However, the location of fuse bits and the electrical characteristics differs between the two devices.

AT90S8535 Compatibility Mode

Programming the S8535C fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 45 for details.
- The double buffering of the USART Receive Register is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 146 for details.





Pin Descriptions

V_{CC}	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	<p>Port A serves as the analog inputs to the A/D Converter.</p> <p>Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega8535 as listed on page 60.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8535 as listed on page 64.</p>
RESET	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 37. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	Input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the Inverting Oscillator amplifier.
AVCC	<p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.</p>
AREF	AREF is the analog reference pin for the A/D Converter.

LAMPIRAN B



PH meter(SKU: SEN0161)



Analog pH Meter Kit SKU: SEN0161



Analog pH Meter Kit SKU: SEN0169

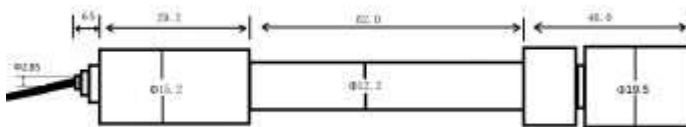
Contents

- ✓ 1 Introduction
- ✓ 2 Specification
- ✓ 3 Precautions
- ✓ 4 pH Electrode Characteristics
- ✓ 5 Usage
 - 5.1 Connecting Diagram
 - 5.2 Method 1. Software Calibration
 - 5.3 Method 2. Hardware Calibration through potentiometer
- ✓ 6 FAQ

Introduction

Need to measure water quality and other parameters but haven't got any low cost pH meter? Find it difficult to use with Arduino? Here comes an analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. You can just connect the pH sensor with BNC connector, and plug the PH2.0 interface into any analog input on Arduino controller to read pH value easily.

Specification



SEN0161 dimension

- ✓ Module Power: 5.00V
- ✓ Circuit Board Size: 43mm×32mm
- ✓ pH Measuring Range: 0-14
- ✓ Measuring Temperature: 0-60 ℃
- ✓ Accuracy: $\pm 0.1\text{pH}$ (25 ℃)
- ✓ Response Time: $\leq 1\text{min}$
- ✓ pH Sensor with BNC Connector
- ✓ PH2.0 Interface (3 foot patch)
- ✓ Gain Adjustment Potentiometer
- ✓ Power Indicator LED

Precautions

- ✓ Before and after use of the pH electrode every time, you need to use (pure)water to clean it.
- ✓ The electrode plug should be kept clean and dry in case of short circuit.
- ✓ **Preservation:** Electrode reference preservation solution is the **3N KCL** solution.
- ✓ Measurement should be avoided staggered pollution between solutions, so as not to affect the accuracy of measurement.
- ✓ Electrode blub or sand core is defiled which will make PTS decline, slow response. So, it should be based on the characteristics of the pollutant, adapted to the cleaning solution, the electrode performance recovery.

- ✓ Electrode when in use, the ceramic sand core and liquid outlet rubber ring should be removed, in order to make salt bridge solution to maintain a certain velocity.

NOTE: Differences between the probes, SEN0161 and SEN0169

Their usages/ specifications are almost the same. The differences locates at

Long-firing Operation: SEN0169 supports, while SEN0161 NOT, i.e. you can not immerse SEN0161 in water for Continuous Testing.

Life Span: In 25 JH, pure water, do Continuous Testing with them both, SEN0169 can work two years, while SEN0161 can only last for 6 months. And just for reference, if put them in turbid, strongly acid and alkali solution, 25JH, the life span would drop to one year (SEN0169), 1 month(or shorter, SEN0161). Temperature, pH, turbidity of the water effect the probe life span a lot.

Waterproof: You can immerse the whole probe SEN0169 into the water, while you can only immerse the front part of the probe SEN0161, the electrode glass bulb, into water, the rear part, from the white shell to the cable, MUST NOT be under water.

Strongly Acid and Alkali: SEN0169 are preferred for strongly acid and alkali test. And if your testing range is usually within pH6~8, then SEN0161 is capable for that.

pH Electrode Characteristics

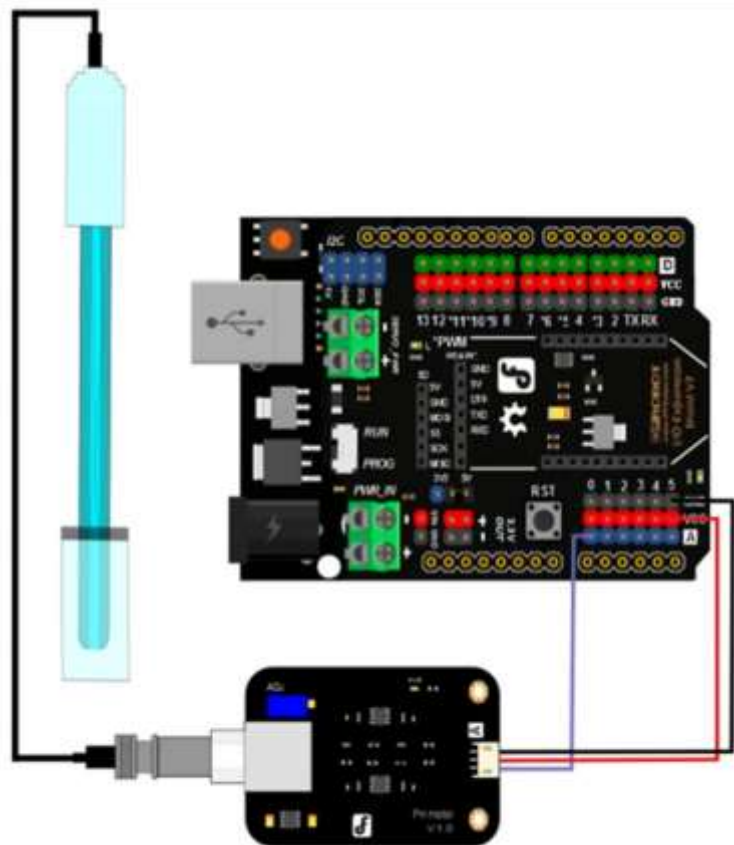
The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 JH):

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

NOTE: It is normal that if your reading is much different with the table since you are not reading from the electrode directly but from the voltage adapter, it has converted the original voltage (-5V ~ +5V) to Arduino compatible voltage, i.e. 0 ~ 5V. [See the discussion on Forum.](#)

Usage

Connecting Diagram



NOTE:

Before you insert the pH probe into one solution from another, or after you finish using the sensor, you must wash the pH electrode with pure water everytime (distilled water is the best)! The closer **power supply** to +5.00V, the more accurate pH readings you could get. You have to immerse the pH probe into stationary solution instead of the running one to get relative stable pH readings. How long should it be under the solution? It depends on the pH value, the closer to neutral solution (pH = 7.00), the longer it will take. As we tested in water pH = 6.0, the blue one costs 6 minutes, and in standard Acid/ Alkali (4.00/ 10.00) solutions, it only needs 10 seconds.

Method 1. Software Calibration

The software calibration is easier than the next part - Hardware Calibration through the Potentiometer. Because it writes the calibration values into Arduino's EEPROM, so you can calibrate once for all if you won't replace your Arduino. It uses mathematical method that to draw a line using two points, i.e. using the Acid standard solution, pH = 4.00 and alkaline pH = 10.00 or 9.18 to draw the linear relation between the voltage and the pH value.



For NOTE 3. Arduino sample sketch
"EEPROM Clear"

NOTE:

During the calibration (from step 4 to step 7), **power outage** should be avoided, or you will have to start over from step 4.

Software Calibration has nothing to do with the **potentiometer** on the adapter. Especially after you finished the calibration, you should never adjust the potentiometer, or you should start over. Moreover, considering the mechanical vibration might interfere the potentiometer value, you could seal it by Hot Melt Adhesive.

If you want to try Hardware Calibration, you'd better reset the EEPROM setting by uploading the Arduino IDE sample sketch **"EEPROM Clear"** as shown as the right hand picture.

Steps

1. Wiring the pH probe, pH meter adapter (the little PCB board) and Arduino UNO as the Diagram section above.
2. Upload the sample code "Software Calibration" below to UNO.
3. Open Serial Monitor, choose command format as "Both NL & CR" and 115200.
4. Send "**Calibration**" to enter Calibration Mode, and you will see "Enter Calibration Mode" directly.

```
calibration
Enter Calibration Mode
Voltage:4262.00mV
Voltage:4262.00mV
```



5. Acid Calibration
 1. Wash your pH probe with pure water (distilled water is best) and dryer it in case of diluting the standard pH solution. Insert it into standard acid solution of pH = 4.0. Wait several seconds till the readings get relative stable.
 2. Enter "**acid:4.00**"(no bland space, lower case), and you will get "Acid Calibration Successful" notice. Then go on with Alkali Calibration.

```
acid:4.00
Acid Calibration Successful
```



6. Alkali Calibration
 1. Take out the pH probe out of the acid solution, CLEAN it again as you did in last step. After this, insert

it into the standard alkali solution with pH = 10 or 9.18.

Waiting for the stable readings

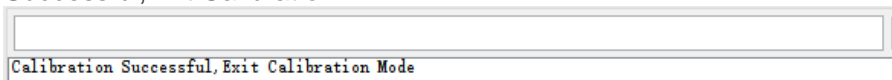
2. Enter **“alkali:10.00”**, and you will see “Alkali Calibration Successful”.



alkali:10.00

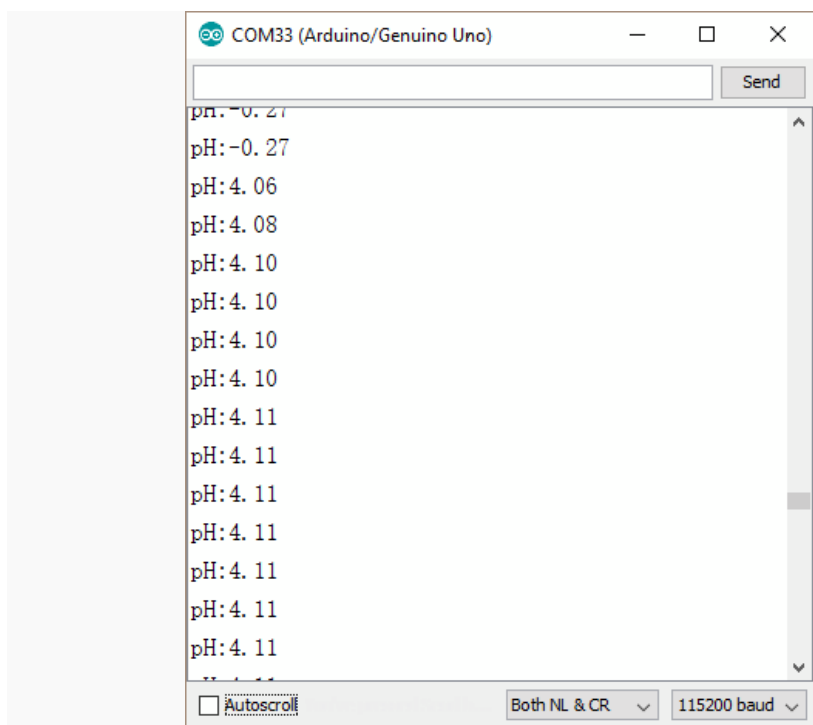
Alkali Calibration Successful

7. Enter “exit” to finish calibration. You will see “Calibration Successful,Exit Calibration Mode”.

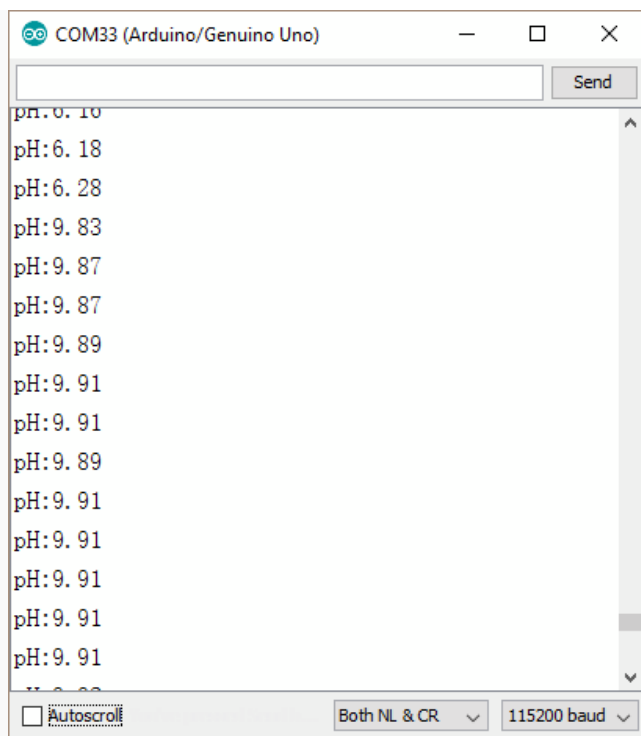


Calibration Successful,Exit Calibration Mode

8. Check if the pH meter was calibrated successfully with the solution pH = 4.00, 9.18, 10.00, if the readings are within the error of 0.1. Congrats!



In Standard acid solution pH
= 4.00



In Standard alkali solution
pH = 10.00

Sample code: Software Calibration

```
/*  
*****
```

```
    This example uses software  
    solution to calibration the ph  
    meter, not the potentiometer. So  
    it is more easy to use and  
    calibrate.
```

```
    This is for SEN0161 and SEN0169.
```

```
    Created 2016-8-11
```

```
    By youyou from DFrobot  
    <youyou.yu@dfrobot.com>
```

```
    GNU Lesser General Public License.
```

```
    See
```

```
    <http://www.gnu.
```

```
    org/licenses/
```

```
    for details. All
```

```
    above must be
```

```
    included in any
```

```
    redistribution
```

```
*****  
*****/
```

```
/*  
*****Notice and  
Troubleshooting*****
```

1.Connection and Diagram can be found here http://www.dfrobot.com/wiki/index.php/PH_meter%28SKU:_SEN0161%29

2.This code is tested on Arduino Uno.

```
*****  
*****/
```

```
#include <EEPROM.h>
```

```
#define EEPROM_write(address, p)  
{int i = 0; byte *pp = (byte*)&(p);for(; i < sizeof(p); i++)  
EEPROM.write(address+i, pp[i]);}
```

```
#define EEPROM_read(address, p)  
{int i = 0; byte *pp = (byte*)&(p);for(; i < sizeof(p); i++)  
pp[i]=EEPROM.read(address+i);}
```

```
#define ReceivedBufferLength 20
```

```
char  
receivedBuffer[ReceivedBufferLength+1]; // store the serial  
command
```

```
byte receivedBufferIndex = 0;
```

```

#define SCOUNT 30 // sum
of sample point

int
analogBuffer[SCOUNT];

//store the sample
voltage int
analogBufferIndex = 0;


#define SlopeValueAddress 0 //
(slope of the ph probe)store at
the beginning of the EEPROM. The
slope is a float number, occupies
4 bytes.

#define InterceptValueAddress
(SlopeValueAddress+4)

float slopeValue, interceptValue,
averageVoltage;

boolean enterCalibrationFlag = 0;


#define SensorPin A0

#define VREF 5000 //for arduino
uno, the ADC reference is the pow
er(AVCC), that is 5000mV


void setup()
{
    Serial.begin(115200);

    readCharacteristicValues();
    //read the slope and intercept of
    the ph probe
}

```

```

void loop()
{
    if(serialDataAvailable() > 0)
    {
        byte modeIndex = uartParse();
        phCalibration(modeIndex);
        // If the correct calibration c
        ommand is received, the calibration
        function should be called.

        EEPROM_read(SlopeValueAddres
s, slopeValue); // After cal
ibration, the new slope and
intercept should be read ,to update
cu rrent value.

        EEPROM_read(InterceptValueAdd
ress, interceptValue);
    }
    static unsigned long
    sampleTimepoint = millis();
    if(millis()-sampleTimepoint>40U)
    {
        sampleTimepoint = millis();

        analogBuffer[analogBufferIndex
] =
        analogRead(SensorPin)/1024.
0*VREF; //read the voltage and
        store into the buffer,every
        40ms analogBufferIndex++;

        if(analogBufferIndex ==
        SCOUNT)analogBufferIndex = 0;

        averageVoltage =
        getMedianNum(analogBuffer, SCOUNT);
    }
}

```



```

// read the stable value by the
median filtering algorithm

    }

    static unsigned long
    printTimepoint = millis();

    if(millis()-
    printTimepoint>1000U)
    {
        printTimepoint = millis();

        if(enterCalibrationFlag)
        // in calibration mode, print the
        voltage to user, to watch the
        stability of voltage

        {Serial.print("Voltage:");
        Serial.print(averageVoltage);
        Serial.println("mV");
        }else{

        Serial.print("pH:");
        // in normal mode, print the ph
        value to user

        Serial.println(averageVoltage/
        1000.0*slopeValue+interceptVal
        u
        e);

        }

    }

}

boolean serialDataAvailable(void)
{

```

```

char receivedChar;

static unsigned long
receivedTimeOut = millis();

while (Serial.available()>0)
{
    if (millis() - receivedTimeOut
        > 1000U)
    {
        receivedBufferIndex = 0;

        memset(receivedBuffer,0,(Rece
            ivedBufferLength+1));
    }

    receivedTimeOut = millis();

    receivedChar = Serial.read();

    if (receivedChar == '\n' ||
        receivedBufferIndex==ReceivedBuffe
        rLength)

```

Method 2. Hardware Calibration through potentiometer

If you've taken the Method 1. Software Calibration, you can ignore this part.

1. Connect according to the graphic, that is, the pH electrode is connected to the BNC connector on the pH meter board, and then use the connection lines, the pH meter board is connected to the analog port 0 of the Arduino controller. When the Arduino controller gets power, you will see the blue LED on board is on.
2. Upload the sample code to the Arduino controller.
3. Put the pH electrode into the standard solution whose pH value is 7.00, or directly short circuit the input of the BNC connector. Open the serial monitor of the Arduino IDE, you can see the pH value printed to it, and the error does not exceed 0.3. Record the pH value printed, then compared with 7.00, and the difference should be changed into the "Offset" in the sample code. For example, the pH value printed is 6.88, so the difference is 0.12. You should change the **# define Offset 0.00** into **# define Offset 0.12** in the sample code.

4. Fine adjustment

For Acid solution: Put the pH electrode into the pH standard solution whose value is 4.00. Then wait about a minute, adjust the Gain Potential device, let the value stabilise at around 4.00. At this time, the acidic calibration has been completed and you can measure the pH value of an acidic solution.

For Alkaline solution: According to the linear characteristics of pH electrode itself, after the above calibration, you can directly measure the pH value of the **alkaline** solution, but if you want to get a better accuracy, you can recalibrate it with the standard solution, pH = 9.18. Also adjust the gain potential device, let the value stabilise at around 9.18. After this calibration, you can measure the pH value of the alkaline solution.

Sample Code for Hardware Calibration

```
/*  
  
# This sample code is used to test the pH  
meter V1.0.  
  
# Editor : YouYou  
  
# Ver      : 1.0  
  
# Product: analog pH meter  
  
# SKU      : SEN0161  
  
*/  
  
#define SensorPin A0           //pH  
meter Analog output to Arduin o Analog  
Input 0  
  
#define Offset 0.00           //deviation  
compensate  
  
#define LED 13  
  
#define samplingInterval 20  
  
#define printInterval 800  
  
#define ArrayLenth 40         //times of  
collection  
  
int pHArray[ArrayLenth];      //Store the  
average value of the sensor feedback  
  
int pHArrayIndex=0;  
  
void setup(void)  
{  
  {pinMode(LED,OUTPUT); Serial.begin(9600);  
    Serial.println("pH meter  
experiment!");    //Test the serial mon  
itor  
  }  
  
  void loop(void){
```

```

static unsigned long
samplingTime =
millis(); static
unsigned long
printTime =
millis(); static
float
pHValue,voltage;
if(millis()-samplingTime >
samplingInterval)
{
    pHArray[pHArrayIndex++]=analogRead(S
ensorPin);

    if(pHArrayIndex==ArrayLenth)pHArrayI
ndex=0;

    voltage = avergearray(pHArray,
ArrayLenth)*5.0/1024;

    pHValue = 3.5*voltage+Offset;

    samplingTime=millis();
}

if(millis() - printTime >
printInterval) //Every 800 milliseco
nds, print a numerical, convert the state
of the LED indicator

{Serial.print("Voltage:");
Serial.print(voltage,2); Serial.print("
pH value: "); Serial.println(pHValue,2);
digitalWrite(LED,digitalRead(LED)^1);
printTime=millis();
}
}

double avergearray(int* arr, int number){

```

```

int i;int max,min; double avg; long
amount=0; if(number<=0){
    Serial.println("Error number for the
    array to avraging!\n");
    return 0;
}
if(number<5){ //less than 5,
    calculated directly statistics
    for(i=0;i<number;i++){
        amount+=arr[i];
    }
    avg = amount/number;
    return avg;}else{
    if(arr[0]<arr[1]){
        min = arr[0];max=arr[1];
    }
    else{
        min=arr[1];max=arr[0];
    }
    for(i=2;i<number;i++){
        if(arr[i]<min){amount+=min;
        //arr<min min=arr[i];
        }else {
            if(arr[i]>max){amount+=max;
            //arr>max max=arr[i];
            }else{
                amount+=arr[i];
                //min<=arr<=max}}//if
            }//for
        avg = (double) amount/ (number-2);
    }//if
    return avg;}

```

FAQ

Q1. My PH sensor readings are not correct, what did I miss?

Or the module is defective?

A. 1. Check if the pH sensor circuit board is good? [Read on the Forum](#). or [on wiki](#) for the steps. During the transport, there might be crash causing the probe head cracked, please check if the probe is good or not.

2. If you don't use Arduino as the controller, then please check your ADC module that whether it converts the 5V analog input to 1024, if it is 4096(or other byte), please re-determine the equation in the code.

Q2. Big fluctuations in ph meter readings. When I make measurements in a glass, I have correct, stable reading. But when I put it inside the aquarium with the pumping system working, the easurement varies even more than a degree, and it's not stable, if I swicth off the pump the given value doesn't oscilate anymore.

A. There should be NO working electrical device in the container. Any tiny leakage of electricity will cause the probe working error. Especially, many people bought the [EC meter](#) and put it into the same tank for the test, but then the pH meter cannot work well anymore. Please seperate them

into different containers, or turning off the EC meter when using the pH meter.

Q3. May I know the Maximum range different if we do not calibrate the pH meter.

A. The maximum range differs from probe, you have to calibrate it before use if the pH probe was kept long.

Q4. I would just like to ask if your pH sensor can be connect to any micro controller aside from arduino. Would it be compatible with a raspberry pi? Thank You!

A. Yes, it can be used on any device as long as it could give 5V power supply and accept 5V analog signal, but as the Rasp pi is only compatible with 3.3V sensor, so an expansion shield is suggested to use with (please make sure which kind of Pi you use)

For any questions and more cool ideas to share, please visit [DFRobot Forum](#)

LAMPIRAN C

/*****

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 7/8/2017

Author : Dimas Gerry A.W

Company :

Comments:

Chip type : ATmega8535

Program type : Application

AVR Core Clock frequency: 8.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 128

/

```
#include <mega8535.h>
```

```
#include <stdio.h>
```

```
#include <delay.h>
```

```
#include <stdlib.h>
```

```
// Alphanumeric LCD functions
```

```
#include <alcd.h>
```

```
#define ADC_VREF_TYPE 0x40
```

```
// Read the AD conversion result
```

```

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

```

// Declare your global variables here

```

void main(void)
{
    // Declare your local variables here
    char lcd_buffer[33];
    char lcd_buffer1[33];
    float vin;
    float in;
    float vin2;
    float ph;
    float in2;
    int i1 = 0;
    int i2 = 0;
    int i3 = 0;
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
    Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T
    State1=T State0=T

    PORTA=0x00;
    DDRA=0x00;

```

```

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
        Func2=Out Func1=In Func0=In
// State7=1 State6=1 State5=0 State4=0 State3=0 State2=0
        State1=T State0=T
PORTB=0xC0;
DDRB=0xFC;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
        Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
        State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
        Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
        State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock

```

```
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
```

```

MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 500.000 kHz
// ADC Voltage Reference: AVCC pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOR&=0xEF;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:

```

```
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);
```

```
while (1)
{
    in = read_adc(1);
    vin = (in * (5.0/1023.0));
    in2 = read_adc(2);
    vin2 = (in2 * (5.0/1023.0));
    ph = (vin2 *3.5);

    lcd_gotoxy(0,0);
    sprintf(lcd_buffer1,"V1: %.2f VOLT ",vin);
    lcd_puts(lcd_buffer1);
    delay_ms(100);
    lcd_gotoxy(0,1);
    sprintf(lcd_buffer,"PH: %.2f ",ph);
    lcd_puts(lcd_buffer);
    delay_ms(100);
```

```
if ( PINB.1 == 1 && i3 == 0 ) // DARI MIKRO 6
{ PORTB.2 = 1 ; // POMPA 2 ON
  // DARI MIKRO 6 DI OFFKAN
  i1 = 1;
  i3 = 1 ;
}
```


[illegible]

[illegible]

```
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);  
delay_ms (100000000);
```

```
PORTB.6 = 1 ;// MOTOR 4 MATI  
PORTB.5 = 0 ; // KE HEATER MATI  
i2 = 1;  
}  
  
}  
}
```

BIODATA PENULIS



Nama Lengkap penulis Dimas Gerry Ainul Wafa' yang dilahirkan di Kota Mojokerto pada tanggal 12 Juni 1995 dari Ayah bernama Junadi dan Ibu bernama Nurul Ayni. Penulis merupakan anak pertama dari dua bersaudara. Saat ini penulis tinggal di Jalan Jayawardhana Mojokerto. Pada tahun 2008, penulis telah menyelesaikan pendidikannya di SDN Kranggan III Mojokerto. Kemudian pada tahun 2011 penulis telah menyelesaikan pendidikan tingkat menengah pertama di SMPN 1

Mojokerto. Tahun 2014 berhasil menyelesaikan pendidikan tingkat menengah atas di SMAN 1 Puri Mojokerto dan melanjutkan studi di Program Studi D3 Metrologi dan Instrumentasi Jurusan Teknik Fisika FTI-ITS. Pada tahun 2015 ini, penulis mampu menyelesaikan Tugas Akhir dengan judul **“RANCANG BANGUN SISTEM PENGENDALIAN pH DALAM PROSES TRANSESTERIFIKASI PADA MINI PLAN BIODIESEL DI WORKSHOP INSTRUMENTASI”**. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi lebih lanjut mengenai Tugas Akhir maka dapat menghubungi penulis melalui email gerryainul@gmail.com.